Predicting a Track and Field Athlete's Race Times: Wayde van Niekerk

David Martinez Vasquez, Giulianna Giordano, and Trey Tipton

Overview

Introduction

In 776 B.C. Coroebus of Elis won the sole event at the inaugural Olympic Games in ancient Greece: the stadion, a 200 meters race the length of the ancient Olympic stadium. By the end of the 6th century B.C., the Olympics had become the most famous of all Greek sporting festivals including events such as discus, javelin, and long jump. The Romans adopted the event and the Olympics thrived until Emperor Theodosius I, a Christian, banned the Games in 394 A.D. due to their ties to paganism. In the mid-1800s, the sport of track and field rekindled in England and eventually gained popularity around the globe.

From July 15 to 24, over 18 million people viewed the 2022 World Athletics Championships across NBC Sports platforms alone. Nearly 150,000 people attended the event in person. Over these ten days, millions of track and field fans watched their favorite athletes succeed or fail to produce results on the world stage.

Question

While some athletes displayed dominance in their respective events or pulled off incredible upsets, others fell short. Such results raise the question: can we predict an athlete's success? More specifically, can we predict an athletes race times based on historical patterns of performance?

In an attempt to answer this question we use data available for a specific athlete, Wayde van Niekerk, from the World Athletics website. The website includes tables displaying Van Niekerk's historical results in the 400-meter dash and other components of the individual races such as the standard of competition and round of competition.

Prior Work

Summary of Prior Work

In 2017, Johannes Hofrichter published an article to his company's website derstatistiker titled Forecast 100m Final Men. In an attempt to predict how the 100 meter race at the 2017 IAAF World Championships might play out, he used historical data from each athlete. Horrichter utilized an unspecified non-linear model to predict race times using wind speed, round of competition (heat, semifinal, or final), and standard of competition (world event or normal event). For example Usain Bolt's model predicted he would run a 9.82 (with some margin of error) in the finals. With the fastest predicted individual time, Hofricter proposed that Bolt would win. Usain Bolt would go on to finish third in the final with a time of 9.95, within the range of the models error.

Critique of Prior Work

While Hofrichter helped show track fans how data science might be used to predict track results, he used only three features in his model: wind speed, round of competition, and standard of competition. However, more variables might influence one's performance on a given day. In their article, Environmental and Venue-related Factors Affecting the Performance of Elite Male Track Athletes, Stephen Hollings et al. suggest altitude,

timing method, and venue type also influence running times. Furthermore, in their article, The Influence of Hot Humid and Hot Dry Environments on Intermittent-sprint Exercise Performance, Mark Hayes et al. suggest temperature and humidity play an important role in running times. Juan Alonso and Jordan Santos-Concejero, Olivier Girard et al., and numerous other echo Hayes et al.'s suggestion. Lastly, running surface might play a role in sprint performance, as suggested by Dominique Stasulli in her article Comparing the Biomechanical Demands of Different Running Surfaces.

Hofrichter could have included a variety of other variables in his predictions. As such, fully trusting his predictions proves difficult.

How This Project Extends Prior Work

Our project aims to extend upon the work of Johannes Hofrichter by incorporating other variables known to influence track and field results. We aim to predict a specific athletes times and, through the process of modeling, determine whether these additional variables help provide better predictions.

Approach

Problem Description

Our project looks specifically at the question, "Can we predict Wayde van Niekerk's race times based on the temperature, date, dew point, atmospheric pressure, track surface, round of competition, and standard of competition of historical races?"

We chose to make predictions using race results from Wayde van Niekerk, the 400 meter dash world record holder from South Africa. In doing so we selected an athlete who continues to compete actively on the professional level and has competed in a specific event (the 400 meter dash) for a long time (since 2012). We chose to use these specific variables based on the those used in Hofrichter models and those thought to contribute to athletics performances.

Approach

To answer this question we will use predictive modeling using historical data concerning Wayde van Niekerk's 400 meter race times. Before doing so we will display some results of our exploratory data analysis (EDA) to provide a sense of how each variable relates to race times. We will create a series of models with the three variables utilized by Hofrichter. We will then incorporate the additional variables into a separate series of models and compare the performance of both groups of models on unseen data.

Provenance

The dataset we use comes mainly from Wayde van Niekerk's profile on the World Athletics website. The profile contains numerous tables with Van Niekerk's historical 400 meter results and includes information such as the date, place, competition name, standard of competition, and round of competition associated with each result. Notably the tables do not include wind speed, as the 400 meter race covers the entirety of the track. Competition organizers provide World Athletics with these competition results. We copied the data from these tables and placed them into a Google Sheets document (we struggled to scrape the data computationally). We then added weather data from Weather Underground, a website which collects historical weather information from 250,000 personal weather stations worldwide. Track surface data was collected from historical IAAF and World Athletics certification documents. In addition, we initially planned on incorporating lane draw data but could not find sufficient information for each race.

Our dataset requires some wrangling. The dates need to be properly formatted and split up for use in our EDA and modeling. We need to refine the round of competition variable (for example, SF1 and SF2 should both just be SF) and the standard of competition variable (to follow Hofrichter's model). We will keep only necessary columns. For example, we do not include place because we are attempting to predict Wayde's results based on the features of the race, not outcomes of the race.

```r
# Load Required Packages
library(tidyverse)
library(lubridate)
library(tidymodels)
#library(rsample)
theme_set(theme_bw())

# Read Dataset
wayde <- read_csv("https://docs.google.com/spreadsheets/d/e/2PACX-1vSaiSPd7hdeFG4u2qPRhB8LE2aJ9bxZd4pCD

# Modify Dataset
wayde <- wayde %>%
  mutate(date = dmy(date)) %>% # properly format date
  mutate(race = ifelse(grepl("SF", race),"SF", # simplify round of competition variable
                       ifelse(grepl("H", race), "H",
                              ifelse(grepl("QF", race), "QF", "F")))) %>%
  mutate(worlds = ifelse(grepl("OW", cat), "Yes", "No")) %>% # simplify standard of competition variabl
  mutate(year = year(date)) %>% # generate new year variable
  mutate(month = month(date)) %>% #generate new month variable
  select(result, date, year, month, worlds, race,# select variables of interest
         temp_avg, atm_pressure, dew_point, certified_track)

# Citation: https://statisticsglobe.com/r-test-if-character-is-in-string
```

Structure

`result`: the duration of the race in seconds

`date`: the date on which the race occurred

`year`: the year in which the race occurred

`month`: the month in which the race occurred

`worlds`: whether the race occurred in the Olympics/World Championships (related to standard of competition)

`race`: whether the race occurred in a heat, semifinal or final (related to round of competition)

`temp_avg`: the average temperature on the day of the race in degrees Fahrenheit (related to temperature)

`atm_pressure`: the atmospheric pressure on the day of the race in inches of mercury (related to altitude)

`dew_point`: the dew point temperature on the day of the race in Fahrenheit (related to humidity)

`certified_track`: whether the race occurred on an IAAF/World Athletics certified track (related to track surface)

Types of Variables:

Character : `date`, `year`, `month`, `worlds`, `race`, and `certified_track`

Double: `result`, `temp_avg`, `atm_pressure`, and `dew_point`

Our dataset contains 82 observations and 10 variables. Each row represents an individual 400 meter result by Wayde van Niekerk. For example, the first observation tells us that on 2012 July 8 Van Niekerk ran in final in a non-worlds event with a time of 46.43 on a track with IAAF certified surfacing, a temperature of around 64 degrees Fahrenheit, and a dew point of around 60 degrees Fahrenheit.

Appropriateness for Task

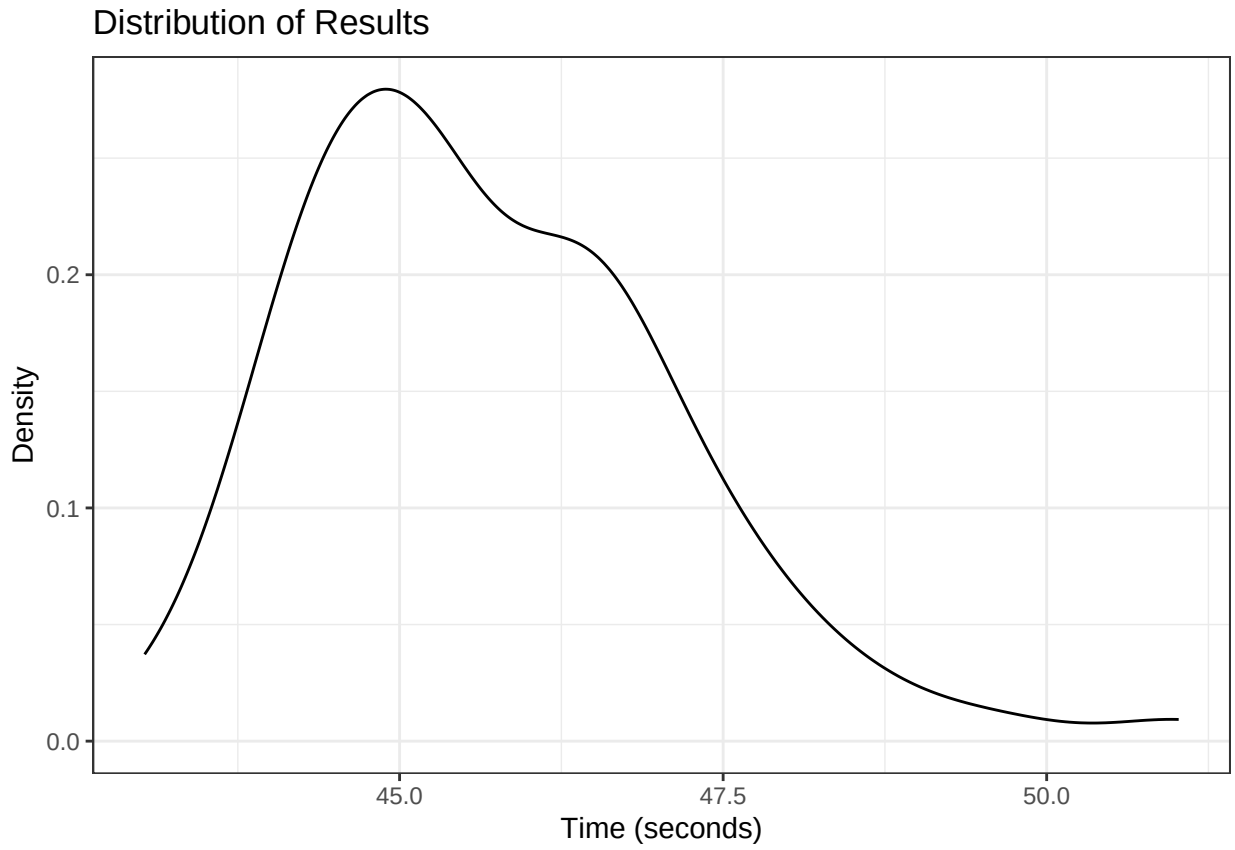
Although our dataset contains many variables, the size (based solely on one athlete) might not prove appropriate for fitting a model. Nevertheless, the features of our dataset seem appropriate to answer our question as we have several variables that affect the speed of runners collectively and individually.

A more appropriate dataset might include information on numerous 400 meter results from various athletes. In addition, additional variables such (specific competitors, lane draw, reaction time, whether an athlete competed in multiple events, etc.) could enhance an optimal dataset. Unfortunately no such dataset exists at this point and would require much effort to create as World Athletics does not publically supply many variables associated with individual races (For more insight into this problem refer to Data-Driven Track & Field/Cross Country by Doug Fenstermacher). As such, we will continue with our dataset acknowledging that it might not provide an appropriate amount of data for our modeling tasks.

Exploratory Data Analysis

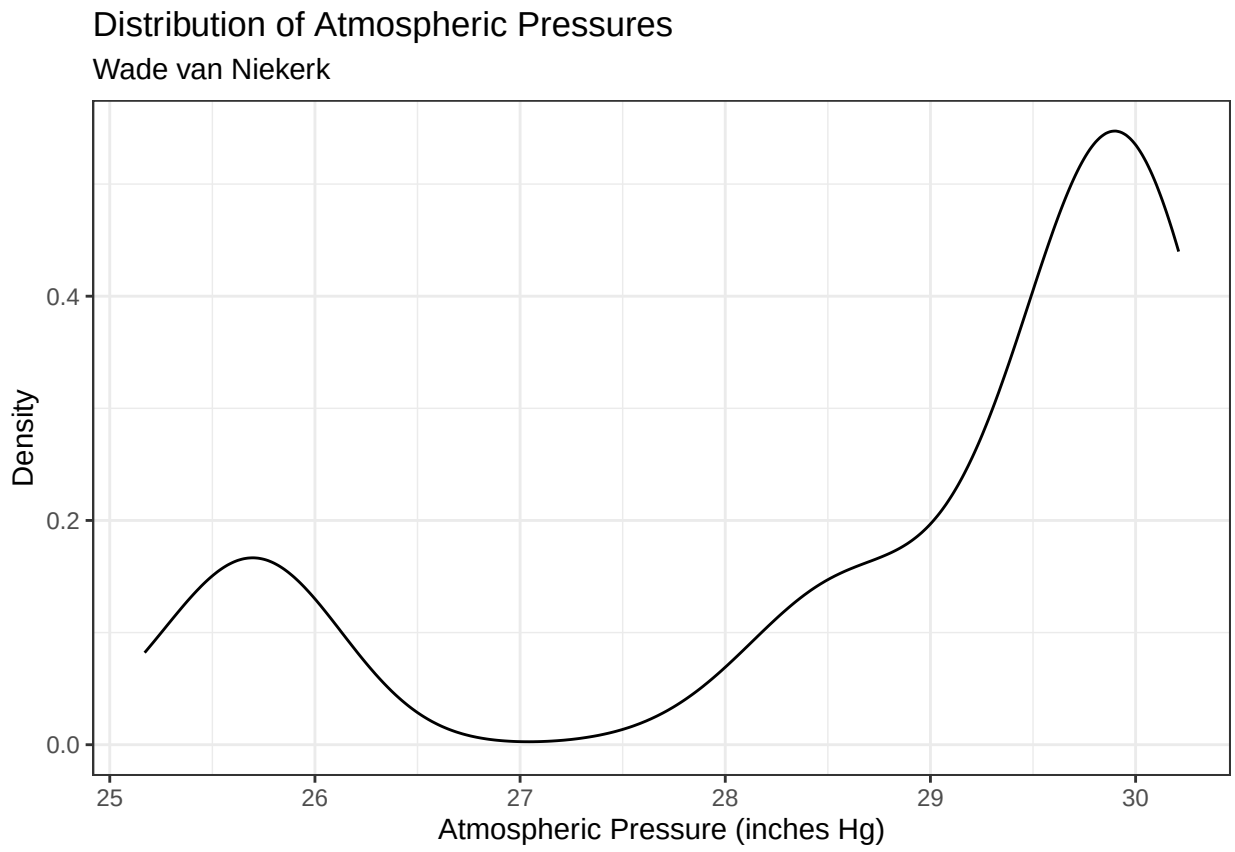
Individual Variables

```
# Exploring the Distribution of Results
wayde %>%
  ggplot(aes(x = result)) +
  geom_density() +
  labs(x = "Time (seconds)", y = "Density", title = "Distribution of Results")
```



This density plot show the distribution of results for Wayde van Niekerk. Van Niekerk has the bulk of his times between 44.5 and 47. The distribution is unimodal, but also has a "bump" indicating more times between 44.5 and 46. The distribution is right skewed.

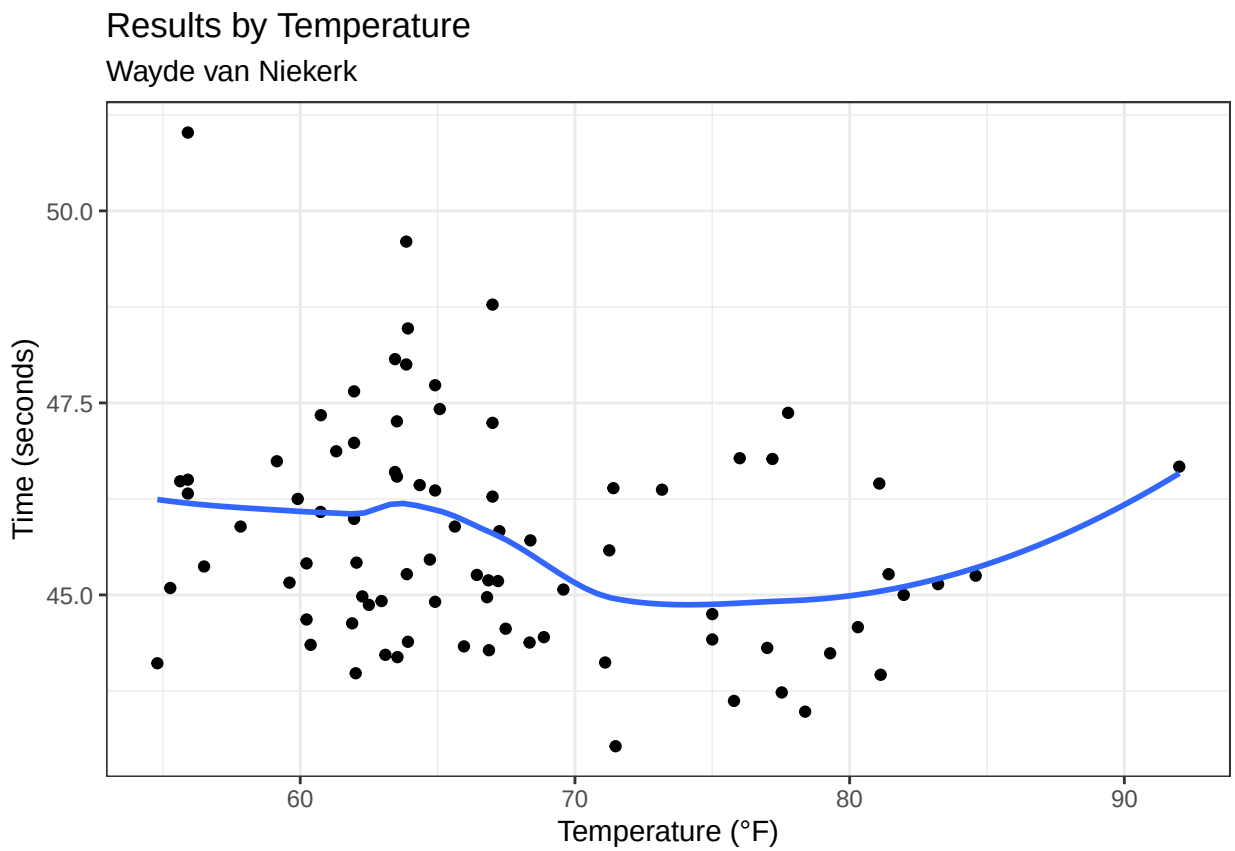
```r
# Exploring the Distribution of Atmospheric Pressure
wayde %>%
  ggplot(aes(x = atm_pressure)) +
  geom_density() +
  labs(x = "Atmospheric Pressure (inches Hg)", y = "Density", title = "Distribution of Atmospheric Pres
```



The distribution for atmospheric pressure at Wayde's races is bimodal with modes at around 25.5 and 29.5.

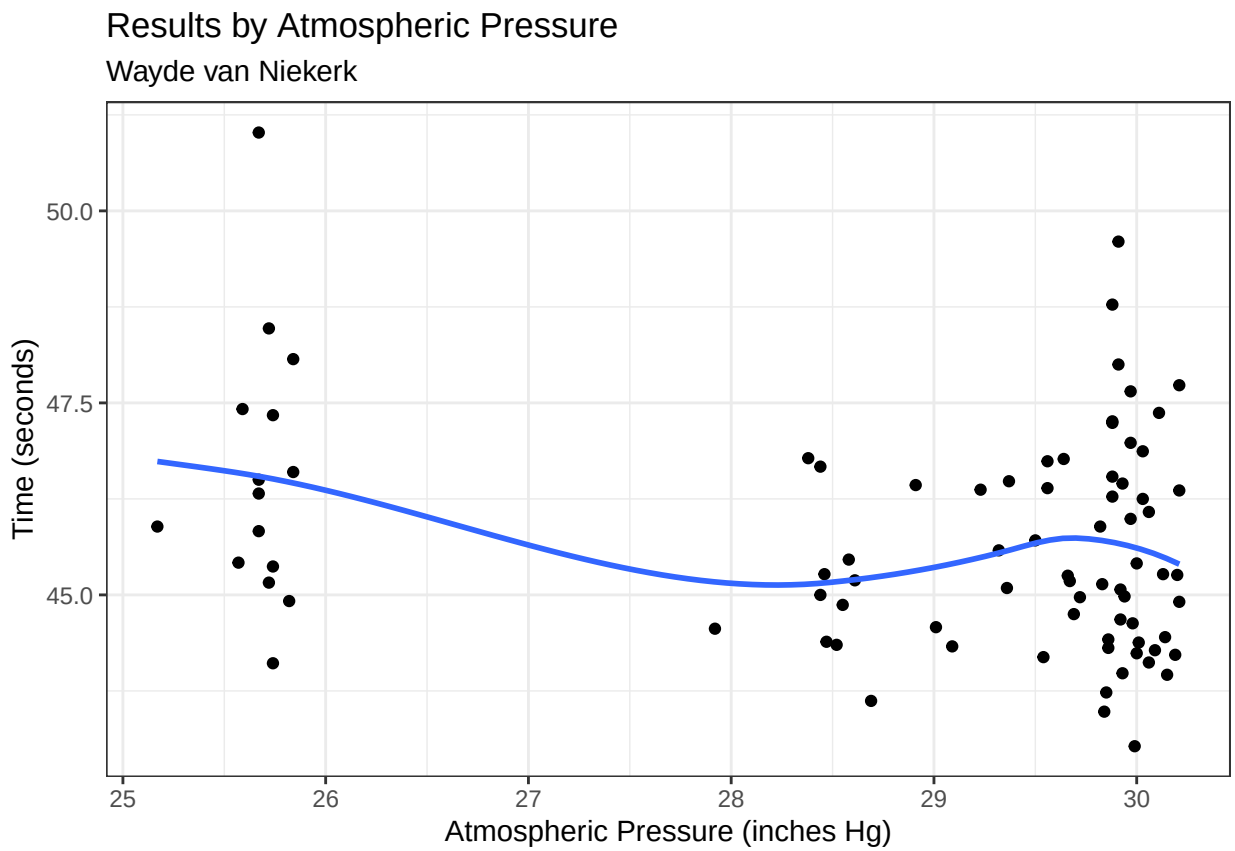
Combinations of Variables

```r
# Exploring Temperature and Results
wayde %>%
  ggplot() +
  aes(x = temp_avg, y = result) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Results by Temperature", x = "Temperature (°F)", y = "Time (seconds)", subtitle = "Wayd
```



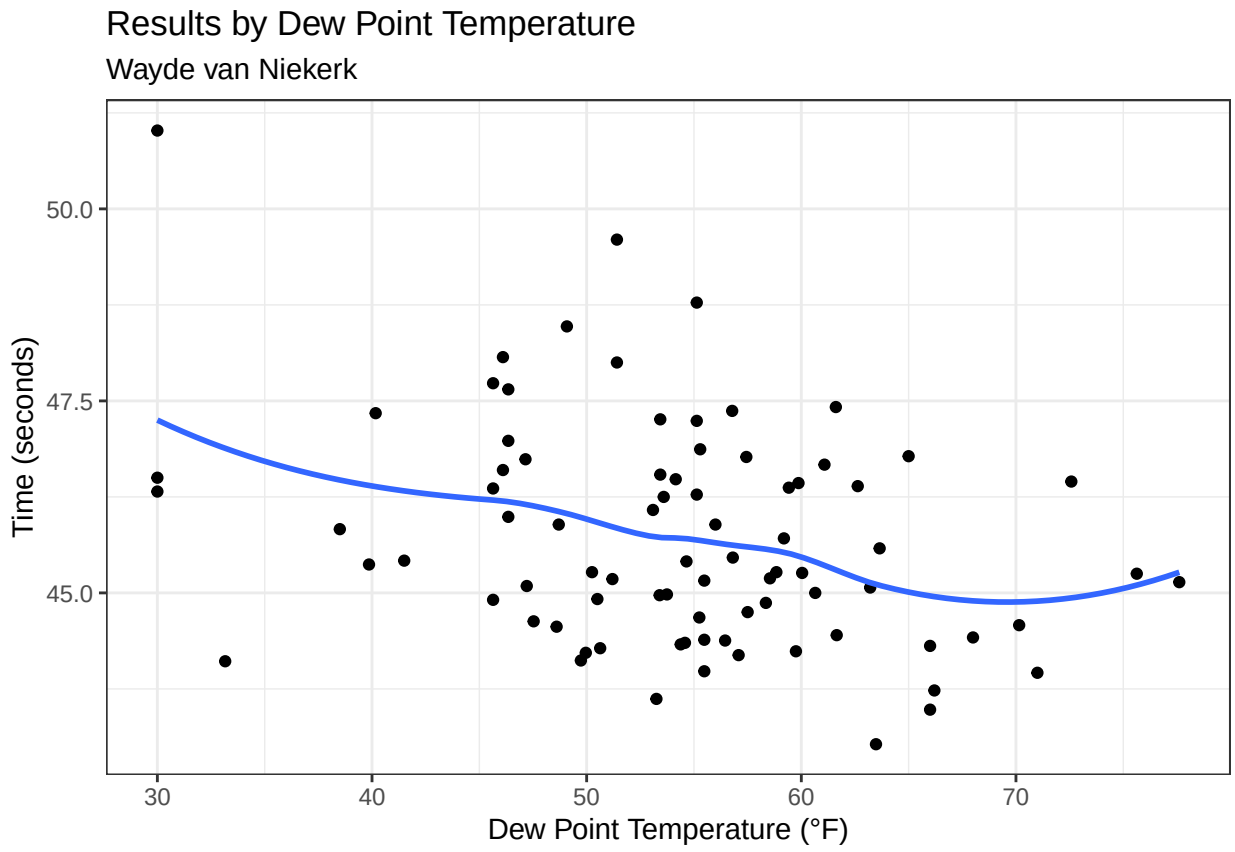
This is a scatterplot of Wayde's results by the average temperature recorded at the event. There does not appear to be an obvious trend, although his times may go slightly down as the temperature goes up. This could be due to warmer weather leading to better races for him.

```
# Exploring Atmospheric Pressure and Results
wayde %>%
  ggplot() +
  aes(x = atm_pressure, y = result) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Results by Atmospheric Pressure", x = "Atmospheric Pressure (inches Hg)", y = "Time (se
```



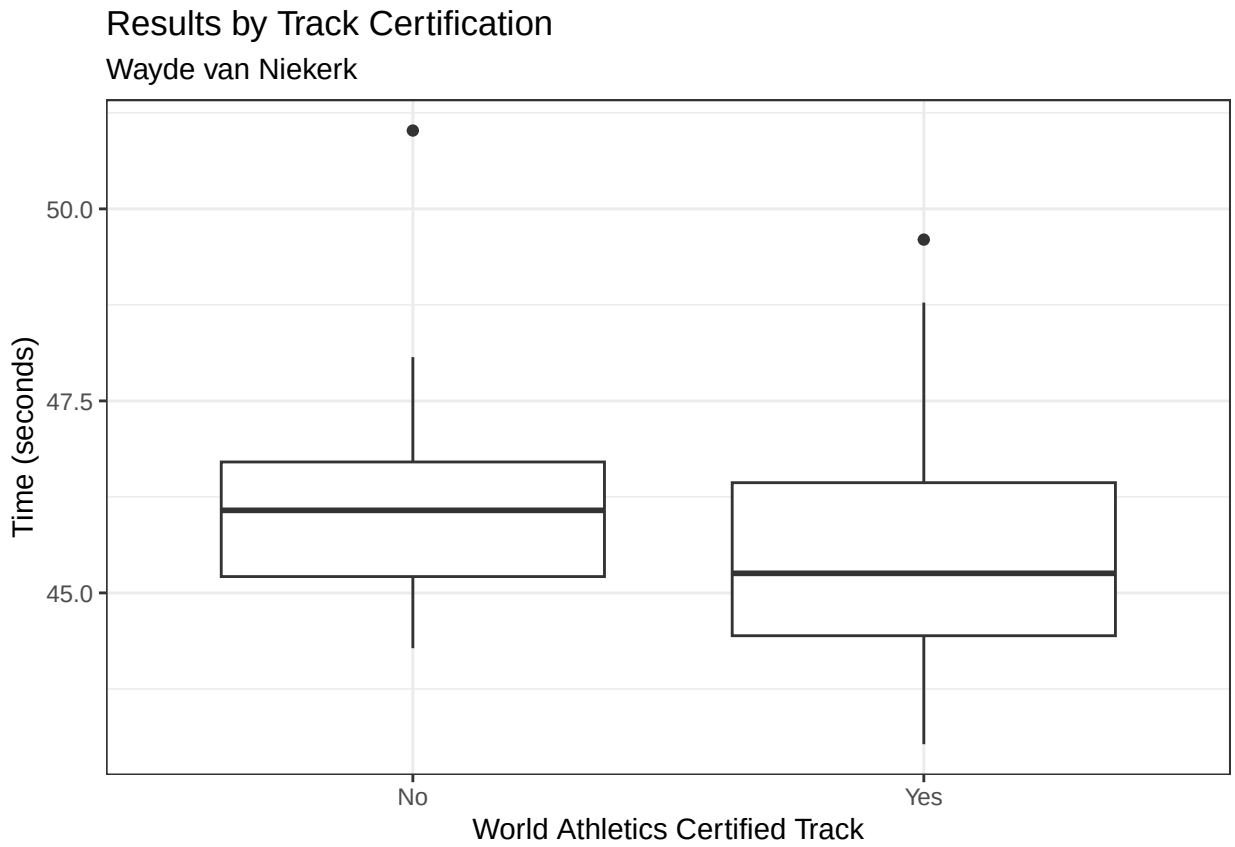
This scatter plot shows the polarity of atmospheric pressure for Wayde's 400 meter races. The trend line reflects that his average times are slightly faster for higher pressures.

```
# Exploring Dew Point Temperature and Results
wayde %>%
  ggplot() +
  aes(x = dew_point, y = result) +
  geom_point() +
  geom_smooth(se = FALSE)  +
  labs(title = "Results by Dew Point Temperature", x = "Dew Point Temperature (°F)", y = "Time (seconds)
```



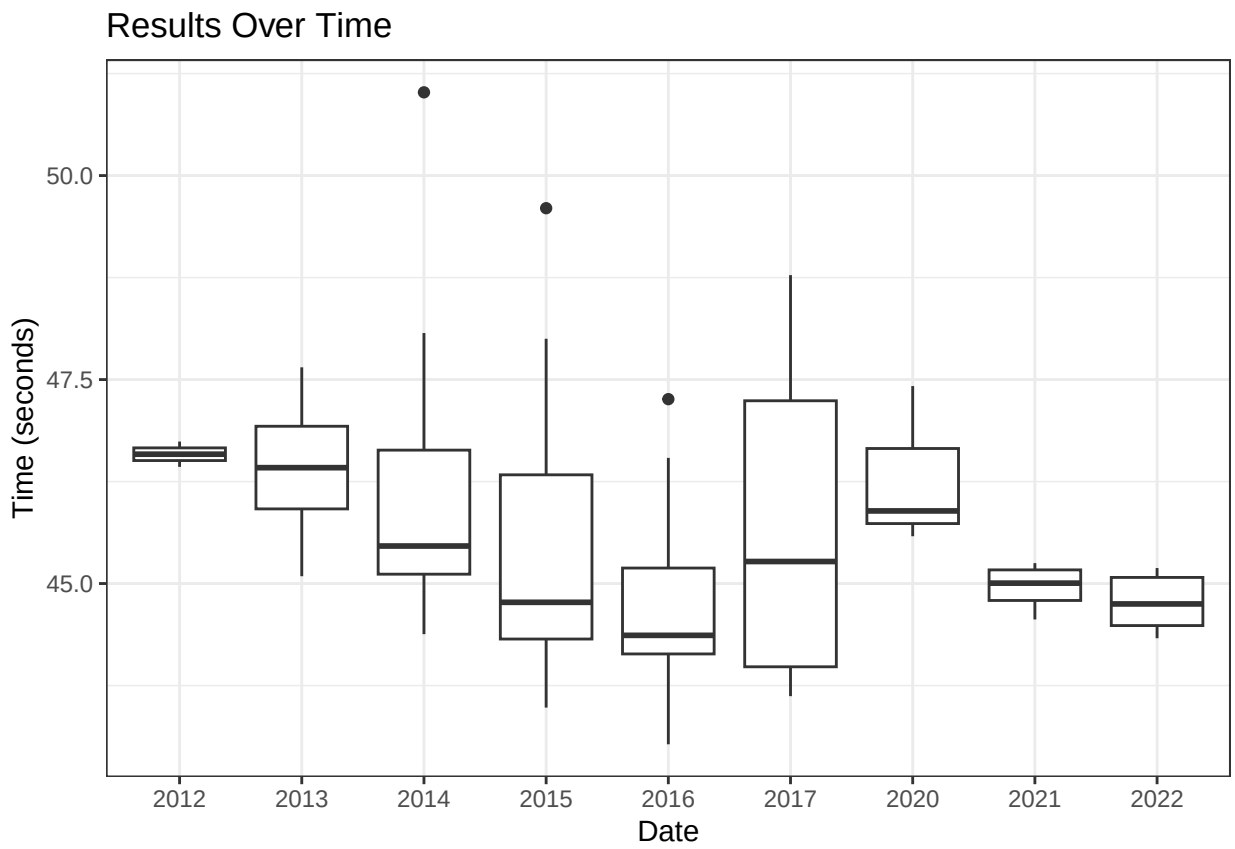
This scatterplot analyzes Wayde's results by the dew point temperature at the race. There seems to be a slight downward trend in Wayde's results as the dew point temperature goes up.

```
# Exploring Track Surface and Results
wayde %>%
  ggplot(aes(x = certified_track, y = result)) +
  geom_boxplot() +
  labs(title = "Results by Track Certification", x = "World Athletics Certified Track", y = "Time (seco
```

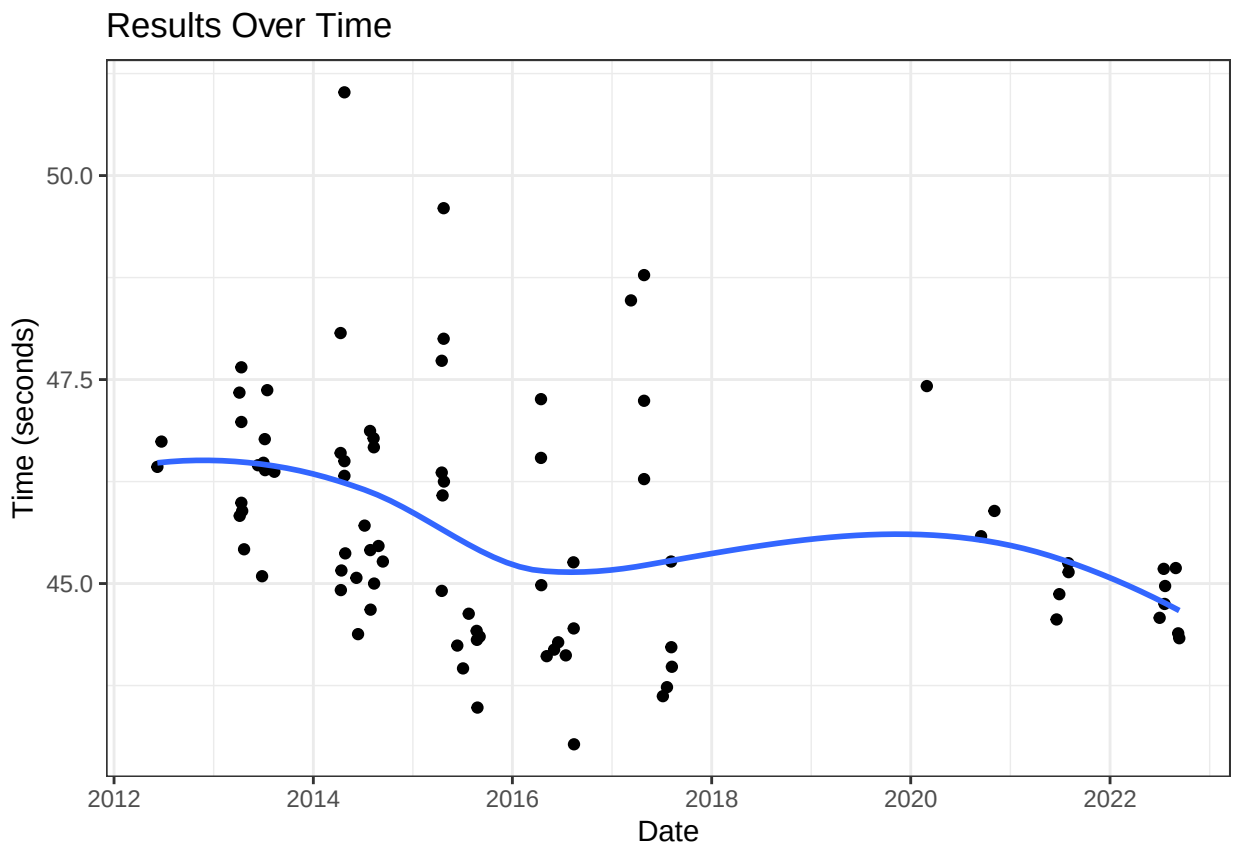


These two box plots compare the distribution of Wayde van Niekerk 400 meter times on tracks that were World Athletics certified and others that were not. Wayde's times appear to be slightly lower on certified tracks though there is overlap between the medians and .

```
# Exploring Results Over Time
wayde %>%
  ggplot(aes(x = as.character(year), y = result)) +
  geom_boxplot() +
  labs(title = "Results Over Time", x = "Date", y = "Time (seconds)")
```

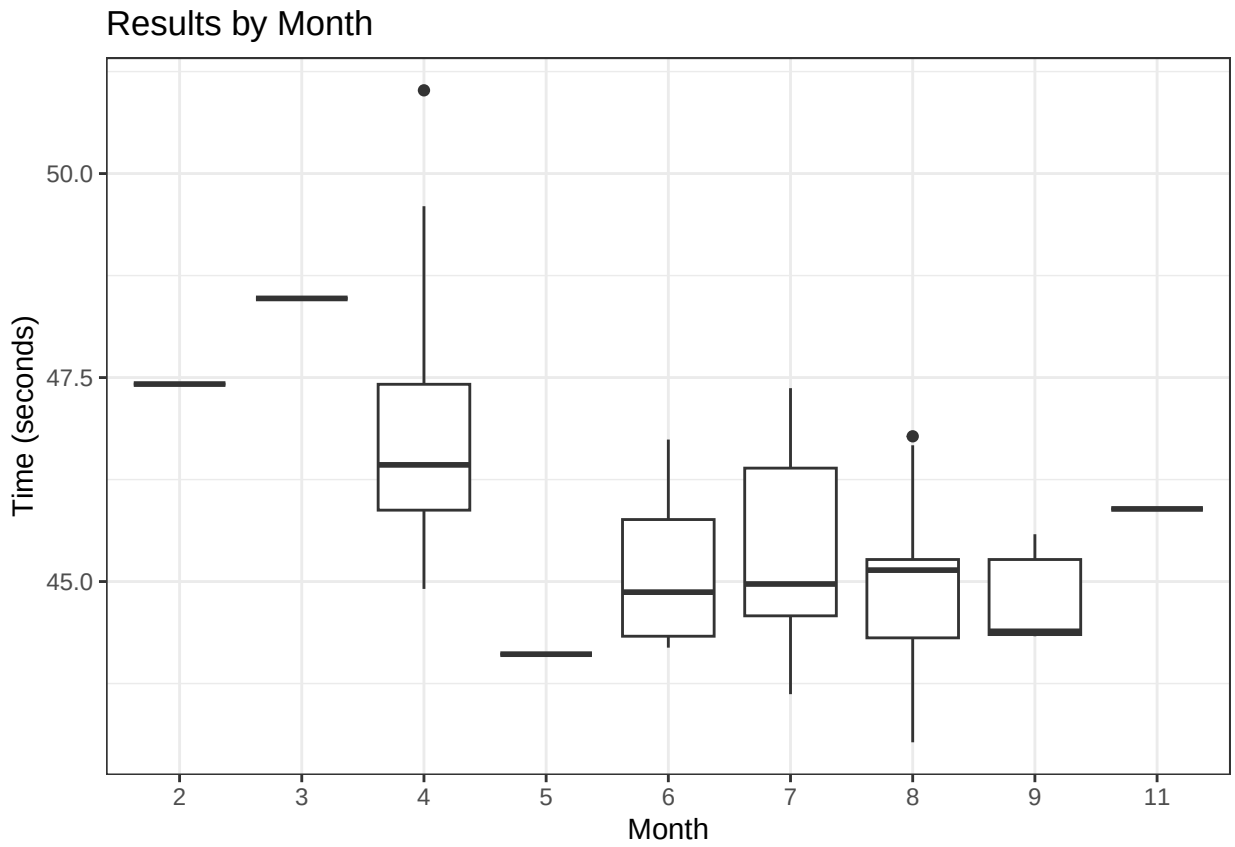



```
wayde %>%
  ggplot(aes(x = date, y = result)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Results Over Time", x = "Date", y = "Time (seconds)")
```



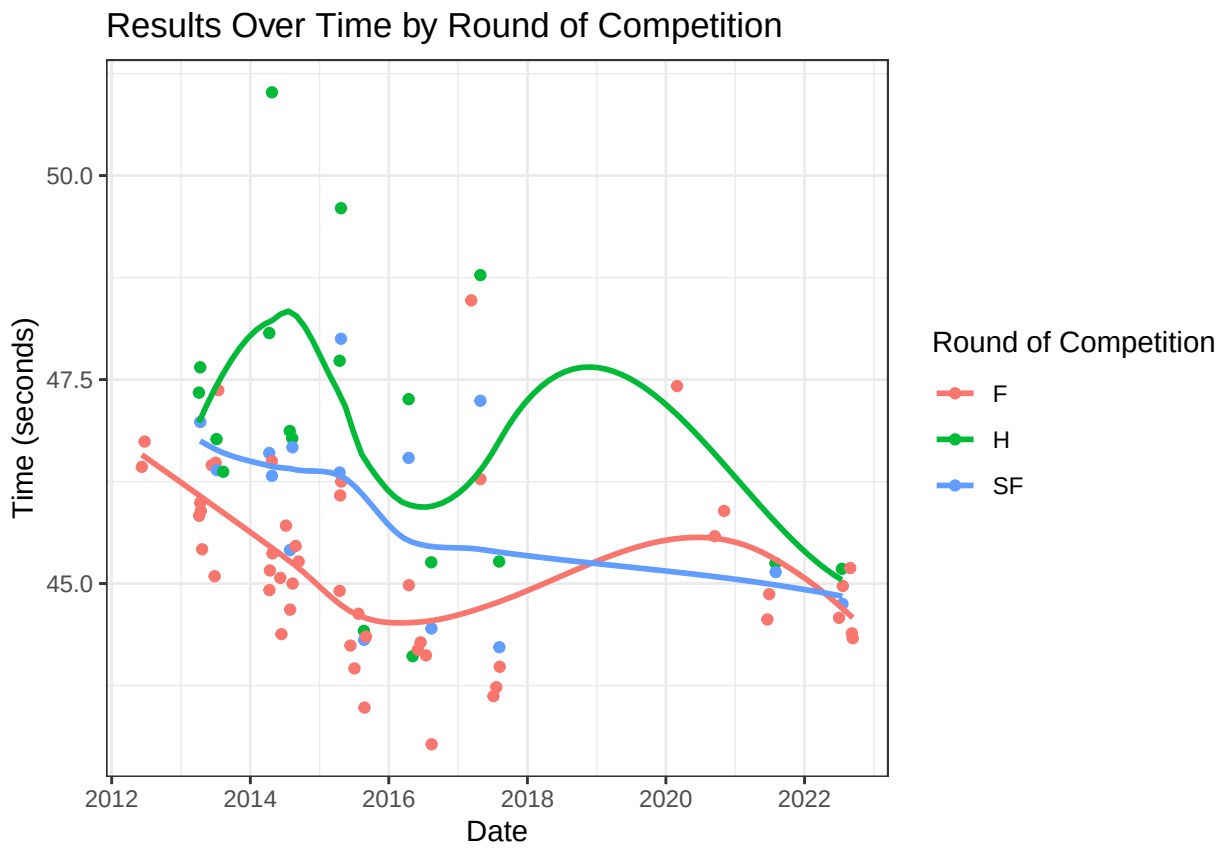
The boxplot and scatterplot analyze Wayde's results in the 400 meter race over time. Wayde's performance seems to mostly improve over the years though some of the medians overlap with other years boxes. Notably, Wayde did not compete during 2018 or 2019. Unfortunately Wayde sustained a knee injury during a game of rugby at the end of the 2017 season (raising further possible issues regarding the appropriateness of our dataset).

```
# Exploring Results by Month
wayde %>%
  ggplot(aes(x = as.factor(month), y = result)) +
  geom_boxplot() +
  labs(title = "Results by Month", x = "Month", y = "Time (seconds)")
```



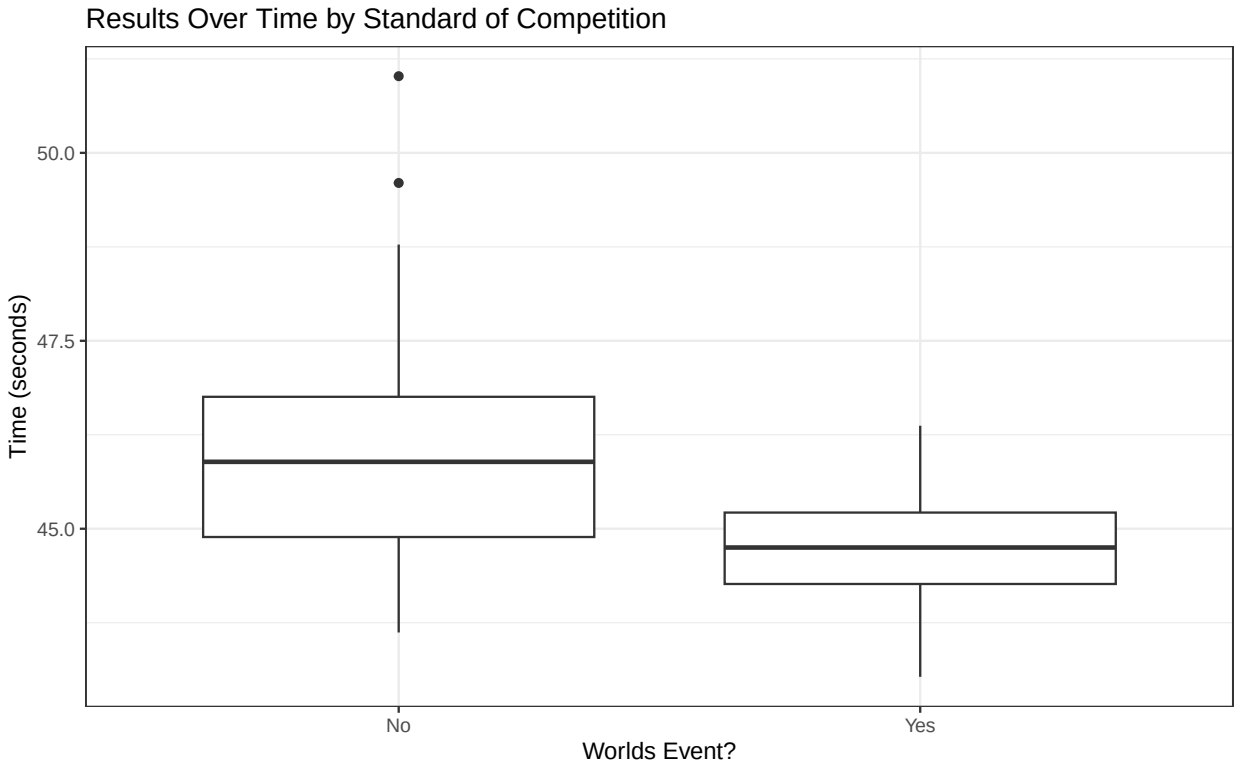
Wayde clearly races less in the winter months, and for some reason, in May. His races in summer months tend to be lower, likely because that is when more important worlds events occur, like the olympics.

```
# Exploring Round of Competition and Results
wayde %>%
  ggplot(aes(x = date, y = result, color = race)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Results Over Time by Round of Competition", x = "Date", y = "Time (seconds)", color = "
```



This is a scatterplot of Wayde's results over time with round of competition taken into consideration. It seems that he performs better in Finals races than other events.

```
# Exploring Event Type and Results
wayde %>%
  ggplot(aes(x = worlds, y = result)) +
  geom_boxplot() +
  labs(title = "Results Over Time by Standard of Competition", x = "Worlds Event?", y = "Time (seconds)
```



Here Wayde van Niekerk's times are plotted based on a whether they occurred at a world-level event or not. Van Niekerk's times, on average, were faster at world-level events.

Summary

Overall, our EDA on the dataset helped to determine the features that might influence Van Niekerk's race results. He seems to perform better in finals and in world-level events likely because he needs to run faster in those scenarios than he does in others to win. He also seems to run faster at higher temperatures and at higher dew points. Wayde's performances also seem to change based on the year. On the other hand, atmospheric pressure, month of the race, and track certification seem to show less obvious or perhaps non-existent relationships with race results. With these insights in mind, we expect our models to make predictions based more heavily on the "more influential" features.

Modeling

Modeling Setup

For all of our models, we will use `result`, the time that it takes Wayde van Niekerk to run a 400 meter race, as the target variable. Since we aim to predict a numerical variable we will use regression.

First, we will fit a series of models with the features related to date, standard of competition, and round of competition (`year`, `month`, `worlds`, and `race`). The use of these variables to reflects the prior work of Johannes Hofrichter. Notably, our EDA indicated that most of these variables might relate to race times. We will then fit another series of models with the addition of the remaining features of our dataset and note whether an improvement in predictions relative to our first series of models occurs.

To measure the accuracy of our model we will use mean absolute error (MAE) and mean absolute percent error (MAPE). MAE provides an output with the same units as our target while MAPE provides a percentage. These metrics will allow us to determine differences in accuracy between and within our model series.

Validation of our models will consist of cross-validation. We will use cross-validation rather than only a train

test split to prevent overfitting and increase the robustness of our performance estimates given that we use a small dataset.

Model Fitting (Part 1)

To begin we will split our dataset into training and testing sets. We will also combine the sets back together to make a data frame we can predict on later.

```
# Split Data Randomly
set.seed(123)

wayde_split <- initial_split(wayde, prop = 3/4)

wayde_train <- training(wayde_split)

wayde_test <- testing(wayde_split)

wayde_with_split_marked <- bind_rows(
  train = wayde_train,
  test = wayde_test,
  .id = "split"
) %>% mutate(split = as_factor(split))
```

Next, we will declare the resamples to use for cross-validation.

```
# Declare Resamples
set.seed(123)

wayde_resamples <- vfold_cv(wayde_train, v = 10)
```

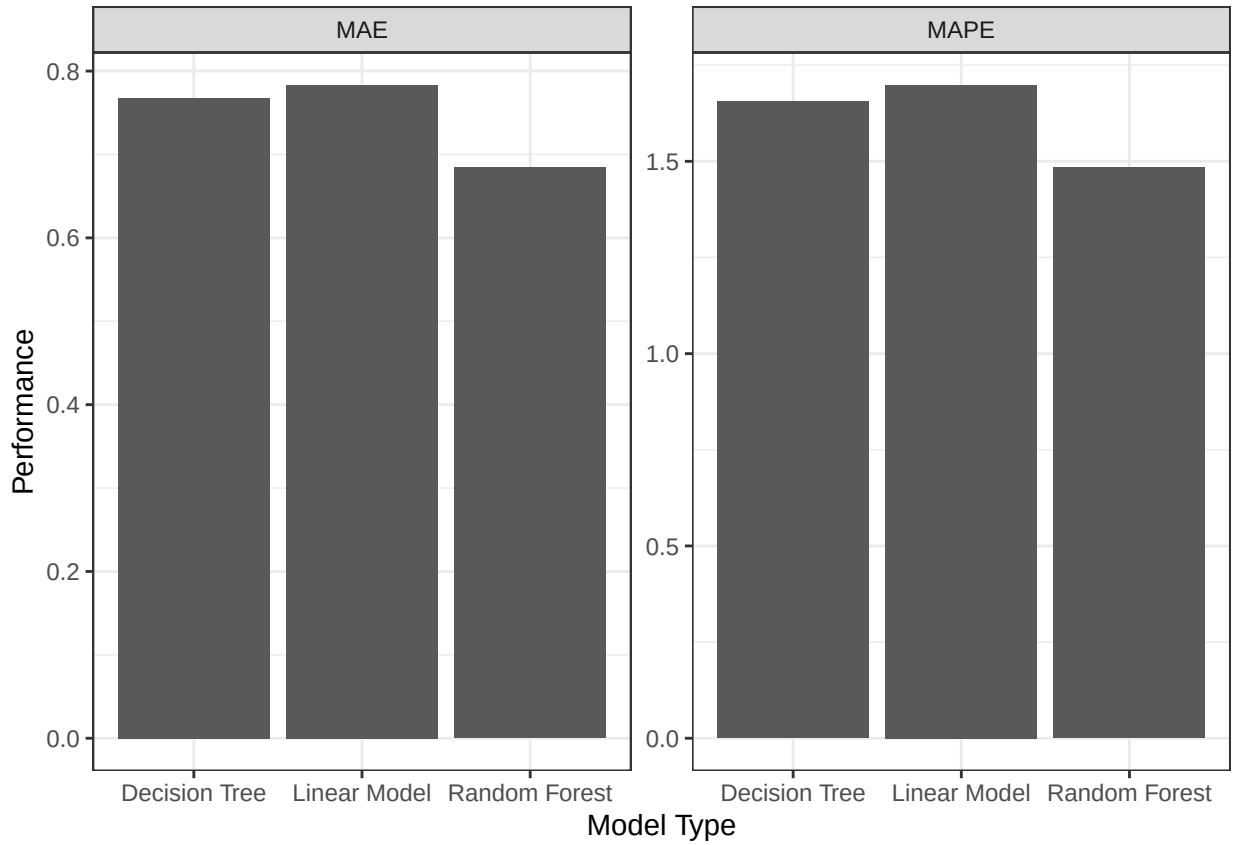
Now we will define our first series of models consisting of a linear, decision tree, random forest, and gradient-boosted tree models. These models are in our semester's "toolkit" for regression tasks.

```
# Define Formula
ms1_recipe <- recipe(result ~ worlds + race + year + month, data = wayde_train)

# Define Models and Workflow
library(parsnip)

lm1 <- workflow(preprocessor = ms1_recipe, spec = linear_reg())
dt1 <- workflow(preprocessor = ms1_recipe, spec =  decision_tree(mode = "regression"))
rf1 <- workflow(preprocessor = ms1_recipe, spec =  rand_forest(mode = "regression"))
bt1 <- workflow(preprocessor = ms1_recipe, spec =  boost_tree(mode = "regression"))
```

Now we will use cross validation to check the model's performance.

```
# Cross-Validation and Metrics
library(tune)
library(yardstick)

lm1_metrics <- fit_resamples(lm1, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  collect_metrics() %>%
  mutate(model = "Linear Model") %>%
  mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             .metric == "mape" ~ "MAPE"))

dt1_metrics <- fit_resamples(dt1, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
```

```
  collect_metrics() %>%
  mutate(model = "Decision Tree") %>%
  mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             .metric == "mape" ~ "MAPE"))

rf1_metrics <- fit_resamples(rf1, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  collect_metrics()  %>%
  mutate(model = "Random Forest") %>%
  mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             .metric == "mape" ~ "MAPE"))

#bt1_metrics <- fit_resamples(bt1, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  #collect_metrics()  %>%
  #mutate(model = "Gradient-Boosted Tree") %>%
  #mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             #.metric == "mape" ~ "MAPE"))

# Display Metrics
bind_rows(
  lm1_metrics,
 dt1_metrics,
  rf1_metrics) %>%
  select(model, .metric, mean, std_err) %>%
  ggplot(aes(x = model, y = mean, group = model)) +
  geom_col() +
  facet_wrap(~.metric, scales = "free_y") +
  labs(x = "Model Type", y = "Performance")
```

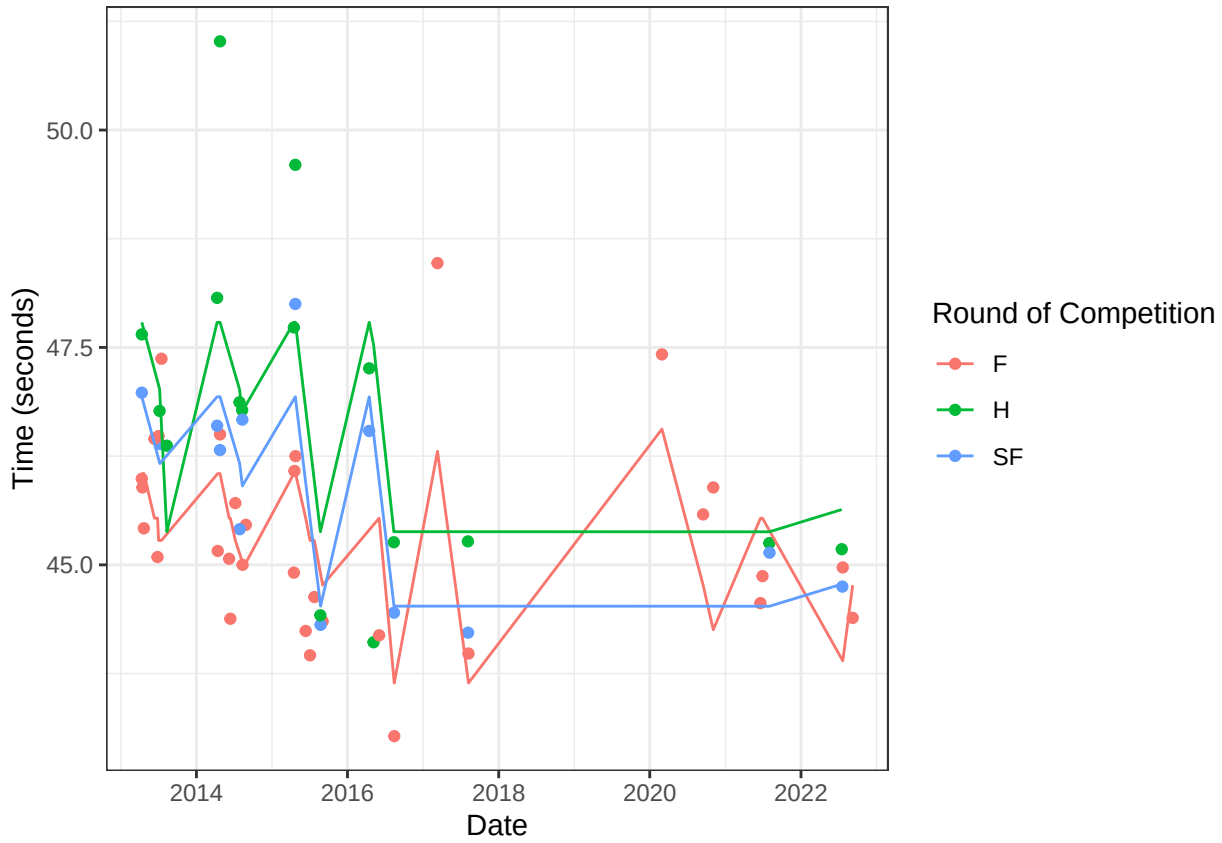


These results indicate that our random forest model might make better predictions on unseen data relative to the decision tree and linear models. For the training data, the linear model makes predictions that are off by about 0.68 seconds (1.5%) on average. Notably, the cross-validation attempt of our gradient-boosted tree model resulted in an error. Now we will evaluate our models using the testing set. First we will fit the models on the training set.

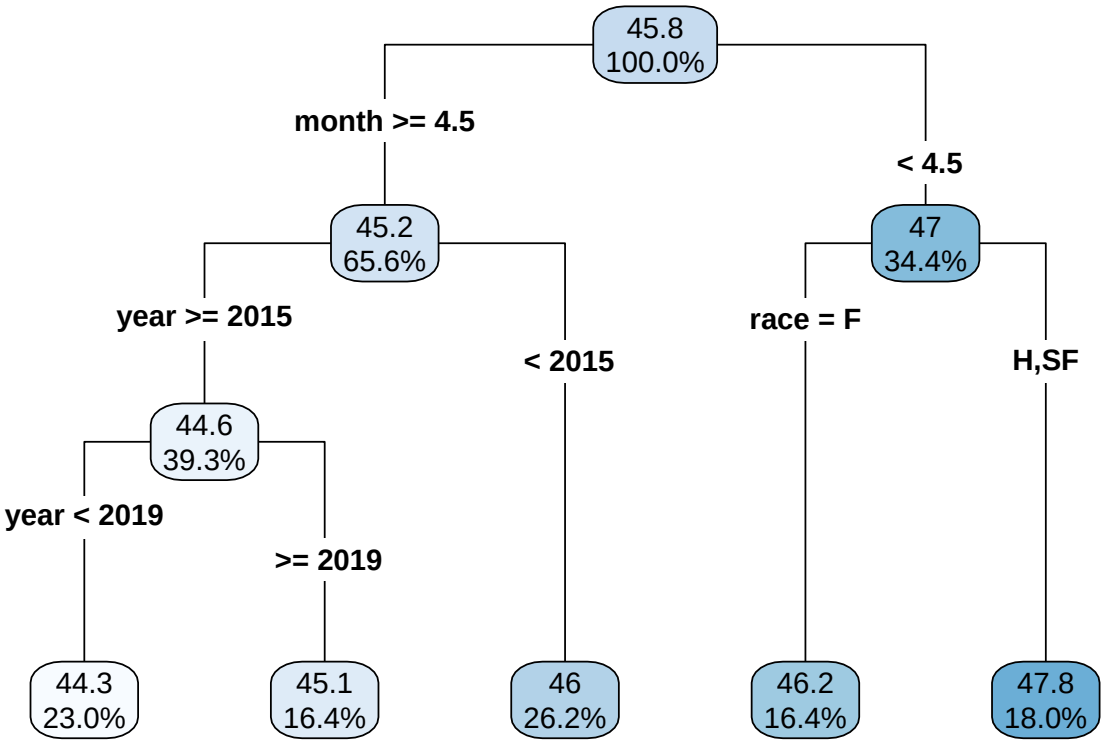
```
 # Define Formula and Fit Models
form1 <- as.formula("result ~ worlds + race + year + month") # OTHER DATE FEATURES TO CONSIDER?

lm1_fit <- fit(linear_reg(), form1, data = wayde_train)
dt1_fit <- fit(decision_tree(mode = "regression"), form1, data = wayde_train)
rf1_fit <- fit(rand_forest(mode = "regression"), form1, data = wayde_train)
bt1_fit <- fit(boost_tree(mode = "regression"), form1, data = wayde_train)
```

We will visualize how the linear and decision tree models make predictions on the training set.

```
# Linear Model Prediction Visualization
augment(lm1_fit, wayde_train) %>%
  ggplot(aes(x = date, y = result, color = race)) +
  geom_point() +
  geom_line(aes(y = .pred)) +
  labs(x = "Date", y = "Time (seconds)", color = "Round of Competition")
```

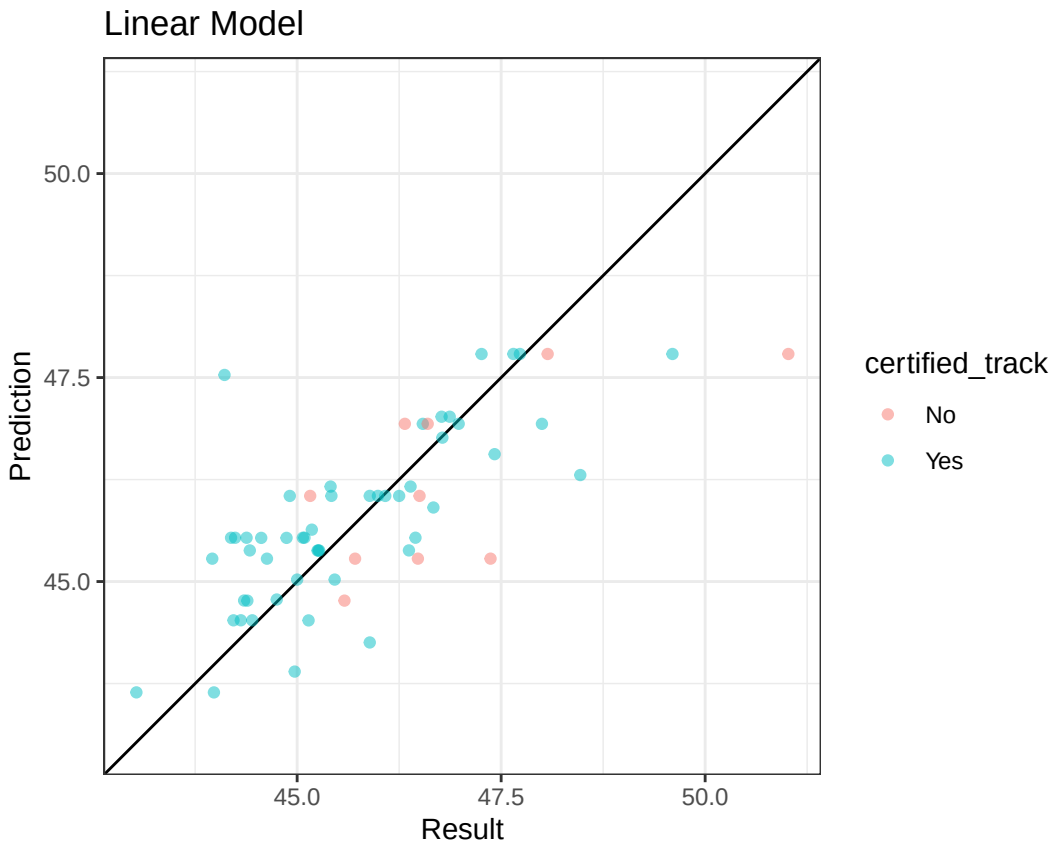



```
# Decision Tree Model Prediction Visualization
library(rpart.plot)
dt1_fit %>%
  extract_fit_engine() %>%
  rpart.plot(roundint = FALSE, digits = 3, type = 4)
```

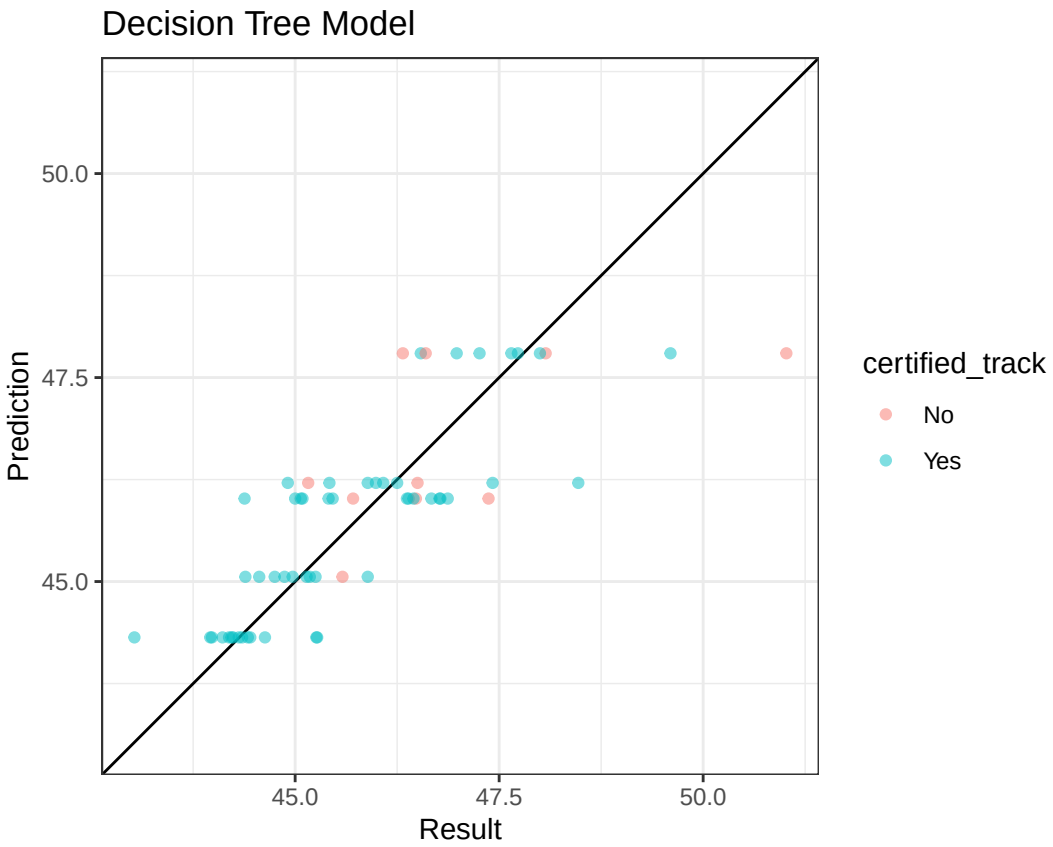


We note that Wayde's time off due to injury has potentially "shaken up" predictions as no data exists for 2018 and 2019. We also see that the decision tree mode uses different features on both sides of tree. We will also create an observed-by-predicted plot for all of our models.

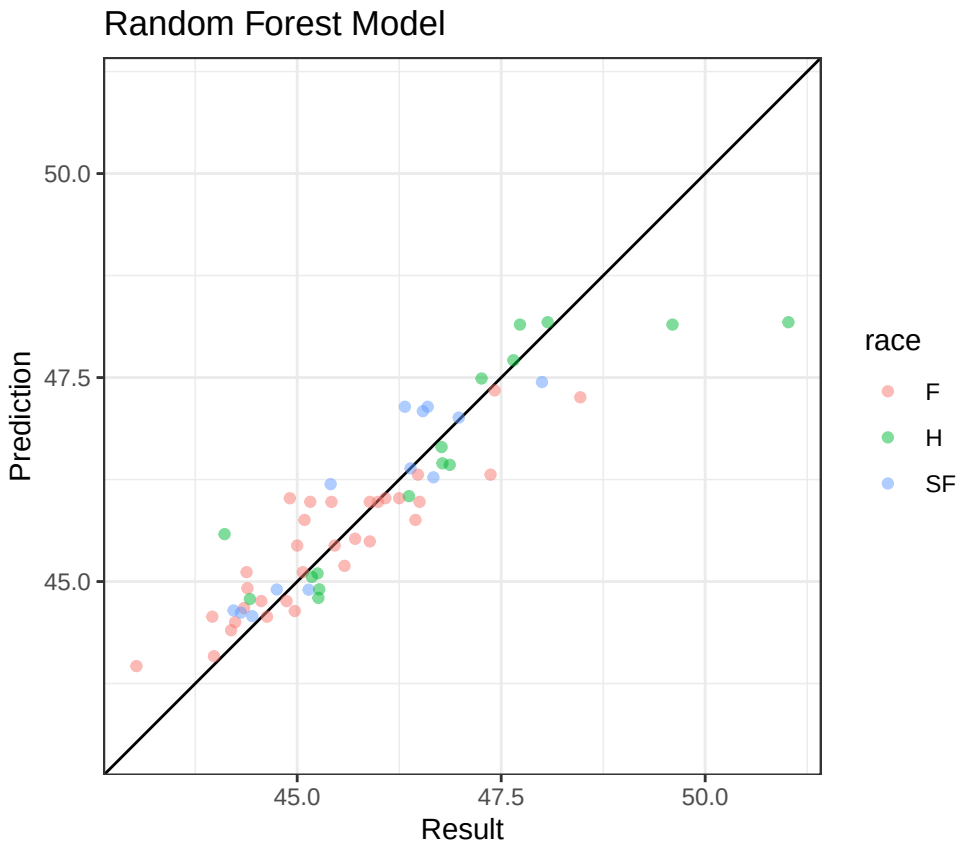
```
show_obs_vs_pred <- function(model, data, var, ...) {
  augment(model, new_data = data) %>%
    ggplot(aes(x = {{var}}, y = .pred, ...)) +
    geom_abline() +
    geom_point(alpha = .5) +
    coord_obs_pred() +
    labs(x = "Result", y = "Prediction")
}

show_obs_vs_pred(lm1_fit, wayde_train, result, color = certified_track) +
  labs(title = "Linear Model")
```



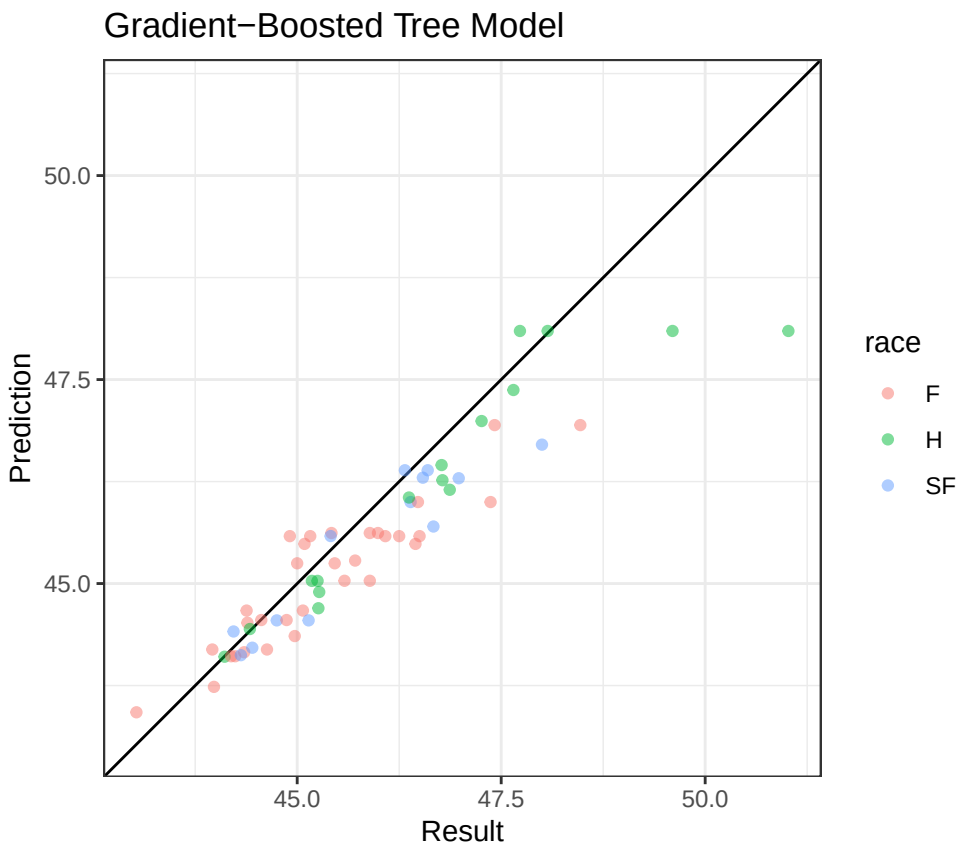

```
show_obs_vs_pred(dt1_fit, wayde_train, result, color = certified_track) +
  labs(title = "Decision Tree Model")
```




```
show_obs_vs_pred(rf1_fit, wayde_train, result, color = race) +
  labs(title = "Random Forest Model")
```




```
show_obs_vs_pred(bt1_fit, wayde_train, result, color = race) +
  labs(title = "Gradient-Boosted Tree Model")
```

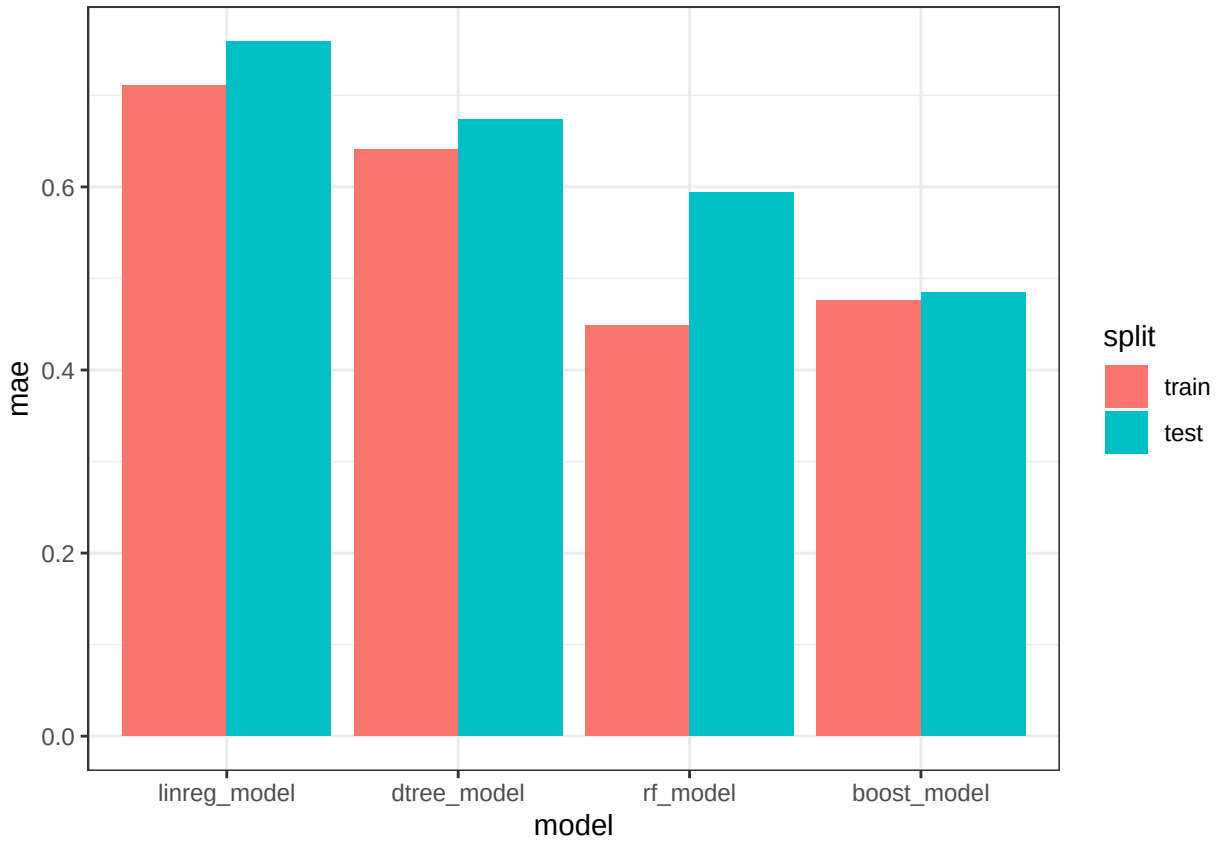



```
# Citation: Prof. Arnold (HW 7)
```

The linear model predicts too low for some of the low temperatures. The decision tree seems to predict too low for some of the earlier years in the dataset. The gradient boosted tree model predicts too high for a lot of values, but especially non final and semifinal events. All of the models seem to predict too high for when there is not a certified track.

Finally, we will quantify the performance of all of our models.

```
eval_dataset <- wayde_with_split_marked

ms1_predictions <- bind_rows(
  linreg_model = augment(lm1_fit, new_data = eval_dataset),
  dtree_model = augment(dt1_fit, new_data = eval_dataset),
  rf_model = augment(rf1_fit, new_data = eval_dataset),
  boost_model = augment(bt1_fit, eval_dataset),
  .id = "model"
) %>% mutate(model = as_factor(model))


ms1_predictions %>%
  group_by(model, split) %>%
  mae(truth = result, estimate = .pred) %>%
  mutate(mae = .estimate) %>%
  ggplot(aes(x = model, y = mae, fill = split)) +
    geom_col(position = "dodge")
```




```
# Citation: Prof. Arnold (HW 7)
```

The best model in terms of prediction performance for the training set was the gradient-boosted tree model. However, for the testing set the best was the random forest model. Notably, the linear regression model was the worst in both instances. The random forest models overfits more than any of the others, as it did well on the training set but not the testing set.

Model Fitting (Part 2)

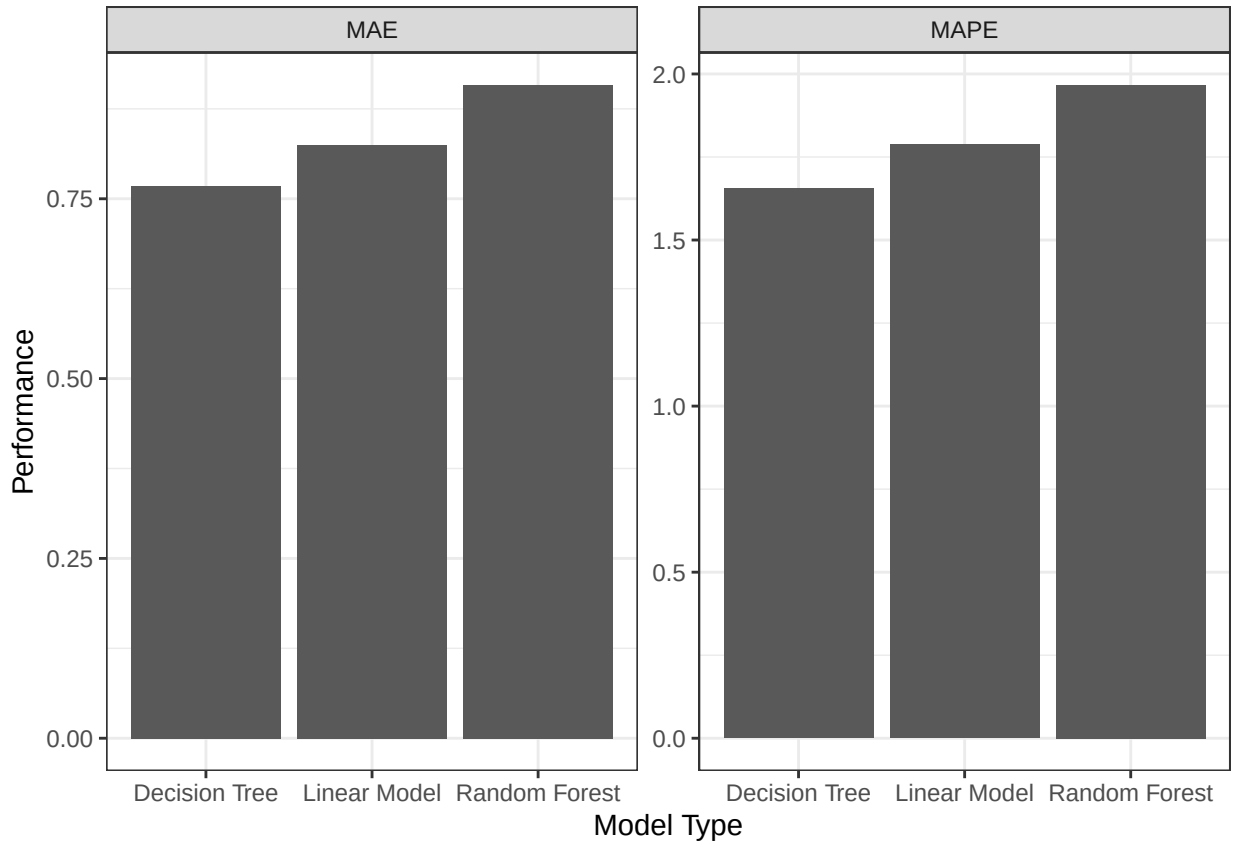
Now we will repeat the previous modeling process but with the a formula that incorporates other features thought to influence race times. These additional features include dew point (a proxy for humidty), average temperature, atmospheric pressure (a proxy for altitude), and track certification (a proxy for track surface).

We will define our second series of models consisting of a linear, decision tree, random forest, and gradient-boosted tree models.

```
# Define Formula
ms2_recipe <- recipe(result ~ worlds + race + year + month + dew_point + atm_pressure + temp_avg + certi

# Define Models and Workflow

lm2 <- workflow(preprocessor = ms2_recipe, spec = linear_reg())
dt2 <- workflow(preprocessor = ms2_recipe, spec =  decision_tree(mode = "regression"))
rf2 <- workflow(preprocessor = ms2_recipe, spec =  rand_forest(mode = "regression"))
bt2 <- workflow(preprocessor = ms2_recipe, spec =  boost_tree(mode = "regression"))
```

We will use cross validation to check the model's performance.

```r
# Cross-Validation and Metrics
lm2_metrics <- fit_resamples(lm2, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  collect_metrics() %>%
  mutate(model = "Linear Model") %>%
  mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             .metric == "mape" ~ "MAPE"))

dt2_metrics <- fit_resamples(dt2, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  collect_metrics() %>%
  mutate(model = "Decision Tree") %>%
  mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             .metric == "mape" ~ "MAPE"))

rf2_metrics <- fit_resamples(rf2, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  collect_metrics()  %>%
  mutate(model = "Random Forest") %>%
  mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             .metric == "mape" ~ "MAPE"))

#bt2_metrics <- fit_resamples(bt2, resamples = wayde_resamples, metrics = metric_set(mae, mape)) %>%
  #collect_metrics()  %>%
  #mutate(model = "Gradient-Boosted Tree") %>%
  #mutate(.metric = case_when(.metric == "mae" ~ "MAE",
                             #.metric == "mape" ~ "MAPE"))

# Display Metrics
bind_rows(
  lm2_metrics,
 dt2_metrics,
  rf2_metrics) %>%
  select(model, .metric, mean, std_err) %>%
  ggplot(aes(x = model, y = mean, group = model)) +
  geom_col() +
  facet_wrap(~.metric, scales = "free_y") +
  labs(x = "Model Type", y = "Performance")
```

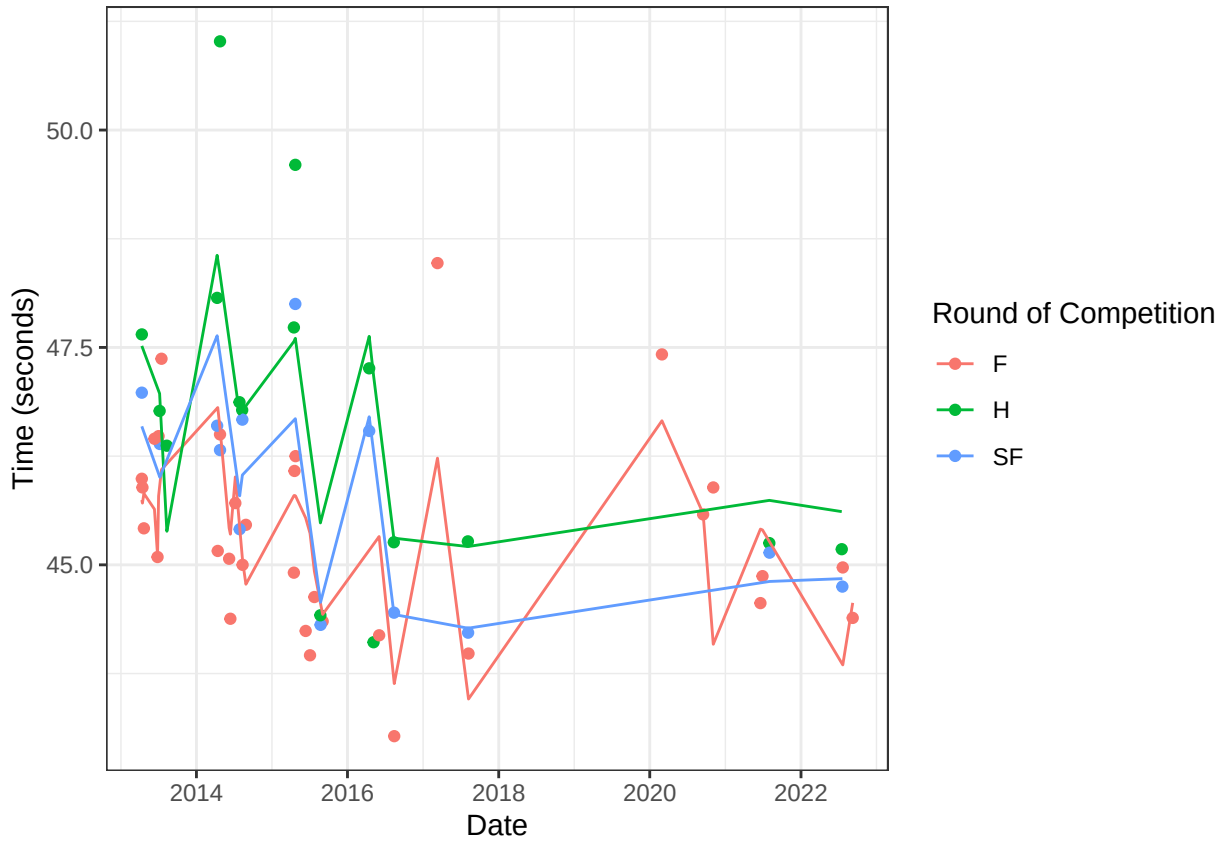


These results indicate that our linear model might make better predictions on unseen data relative to the random forest and decision tree models. For the training data, the linear model makes predictions that are off by about 0.89 seconds (1.9%) on average. Note that this is different than our previous model series. Now we will evaluate our models using the testing set. First we will fit the models on the training set.

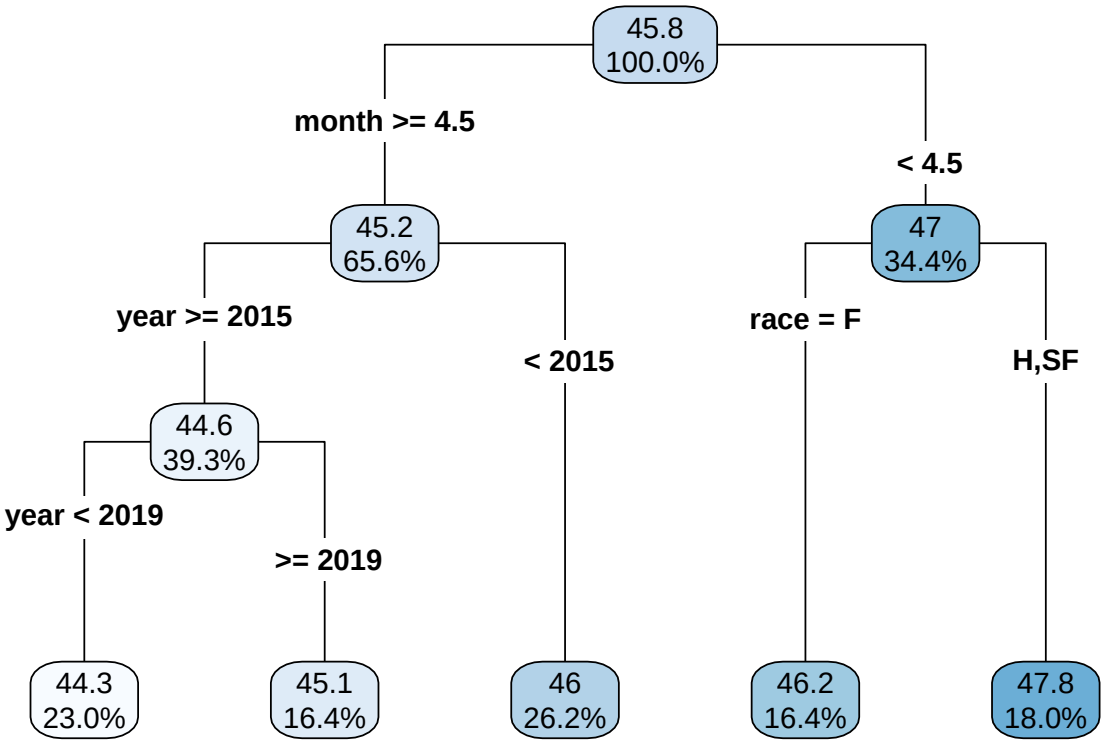
```
 # Define Formula and Fit Models
form1 <- as.formula("result ~ worlds + race + year + month + dew_point + atm_pressure + temp_avg + certi

lm2_fit <- fit(linear_reg(), form1, data = wayde_train)
dt2_fit <- fit(decision_tree(mode = "regression"), form1, data = wayde_train)
rf2_fit <- fit(rand_forest(mode = "regression"), form1, data = wayde_train)
bt2_fit <- fit(boost_tree(mode = "regression"), form1, data = wayde_train)
```

We will once again create an observed-by-predicted plot for all of our models.

```
# Linear Model Prediction Visualization
augment(lm2_fit, wayde_train) %>%
  ggplot(aes(x = date, y = result, color = race)) +
  geom_point() +
  geom_line(aes(y = .pred)) +
  labs(x = "Date", y = "Time (seconds)", color = "Round of Competition")
```



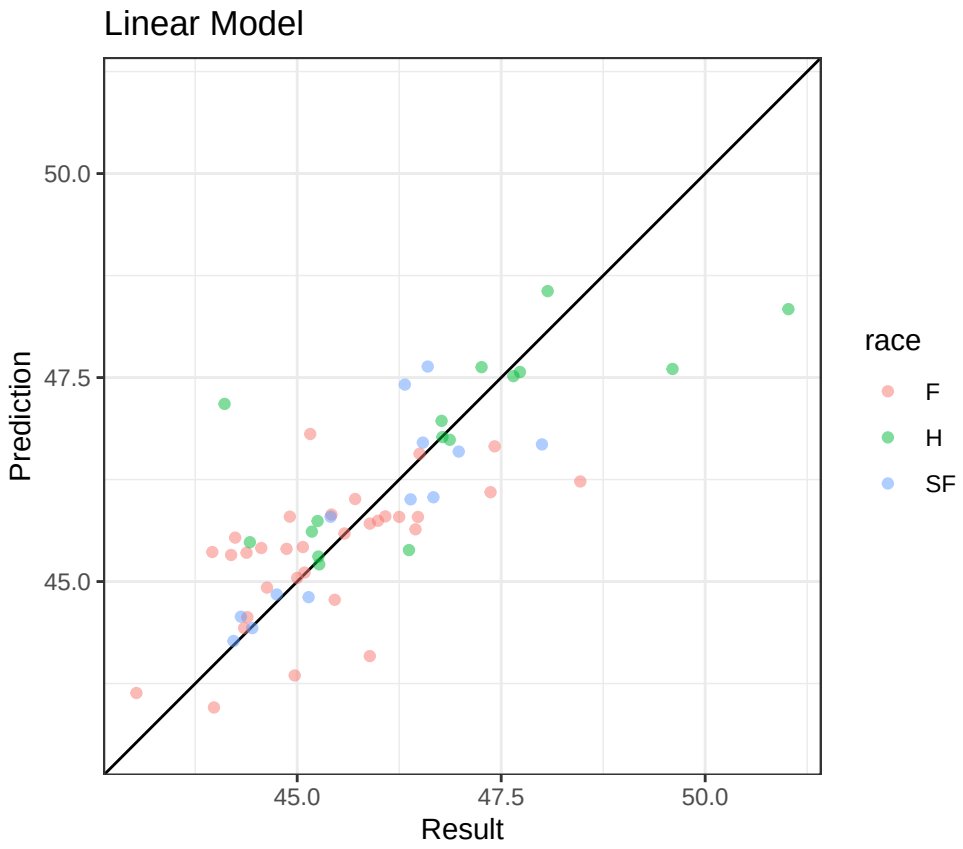

```
# Decision Tree Model Prediction Visualization
library(rpart.plot)
dt2_fit %>%
  extract_fit_engine() %>%
  rpart.plot(roundint = FALSE, digits = 3, type = 4)
```



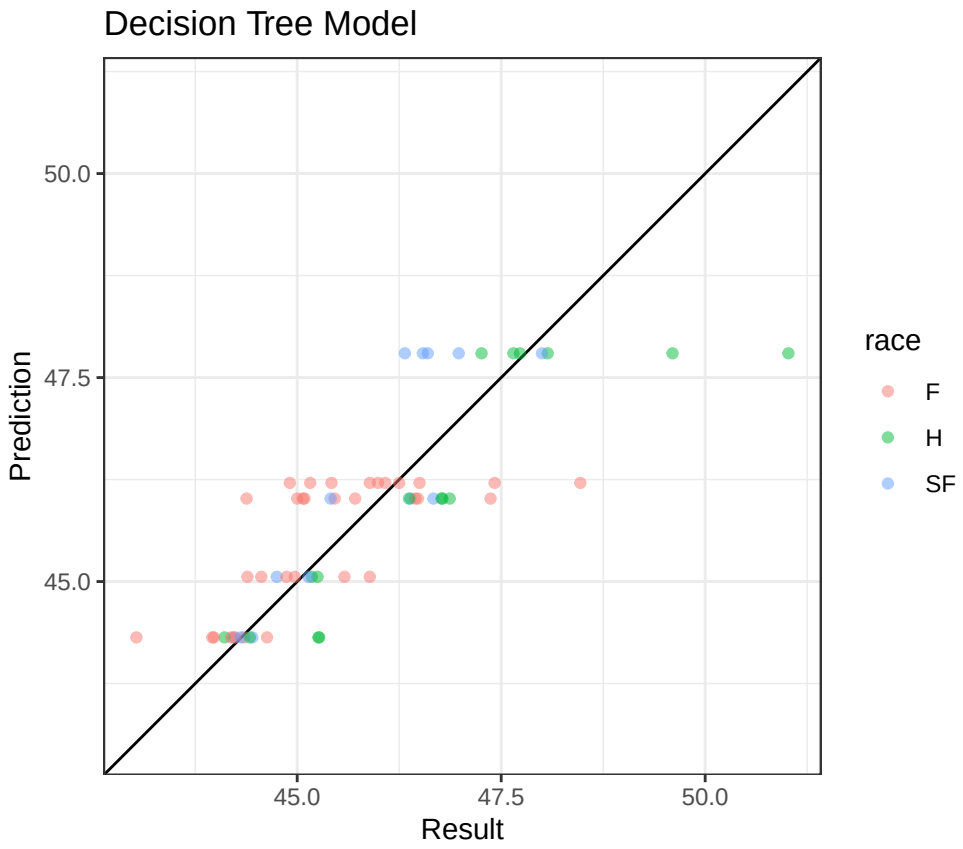
This decision tree model is the exact same as the previous decision tree, so adding in new varibales did not change the results of this model.

We will also create an observed-by-predicted plot for all of our models.

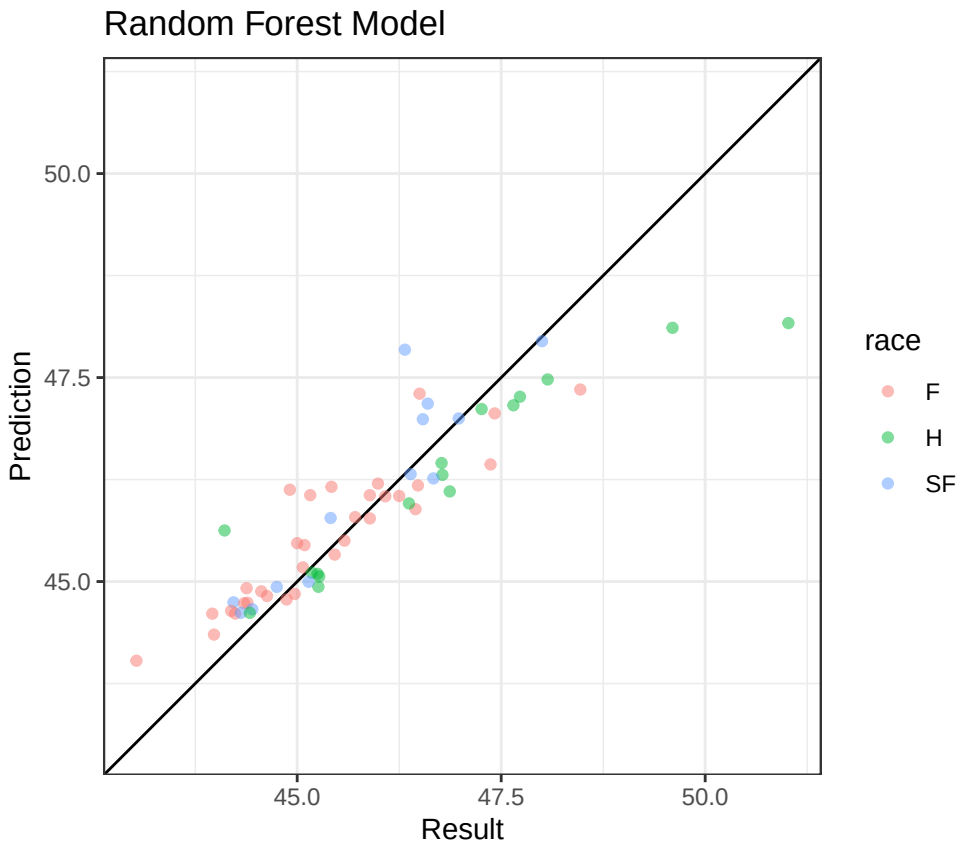
```
show_obs_vs_pred(lm2_fit, wayde_train, result, color = race) +
  labs(title = "Linear Model")
```



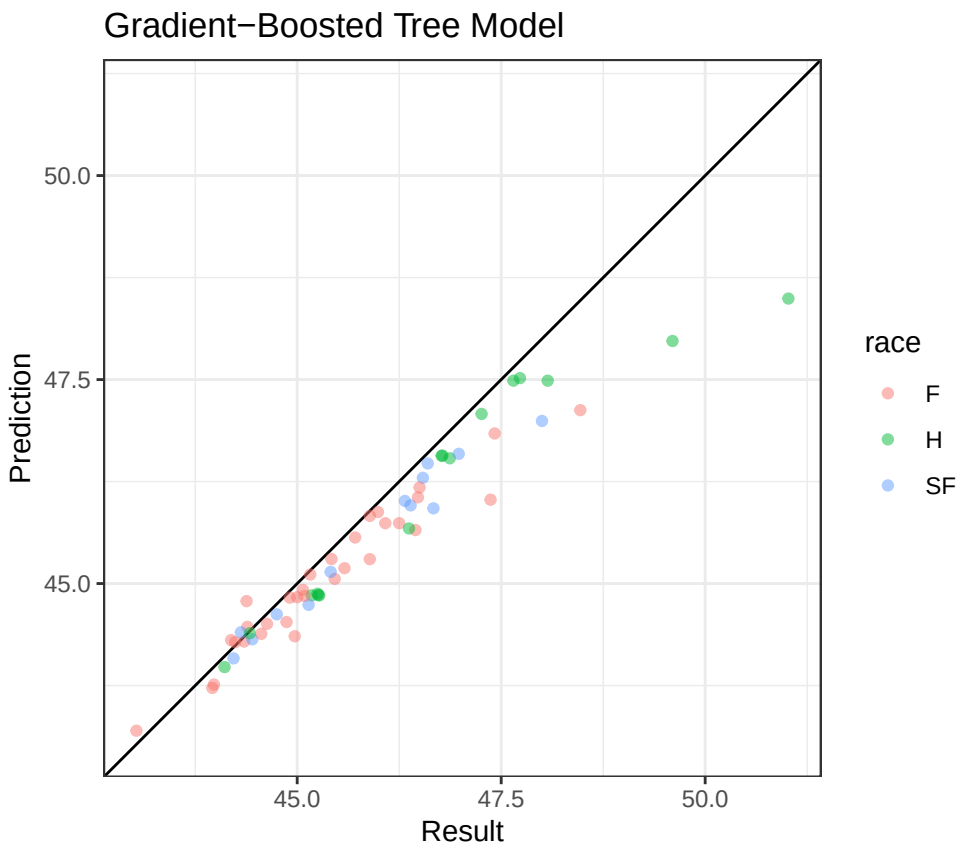

```
show_obs_vs_pred(dt2_fit, wayde_train, result, color = race) +
  labs(title = "Decision Tree Model")
```




```
show_obs_vs_pred(rf2_fit, wayde_train, result, color = race) +
  labs(title = "Random Forest Model")
```




```
show_obs_vs_pred(bt2_fit, wayde_train, result, color = race) +
  labs(title = "Gradient-Boosted Tree Model")
```

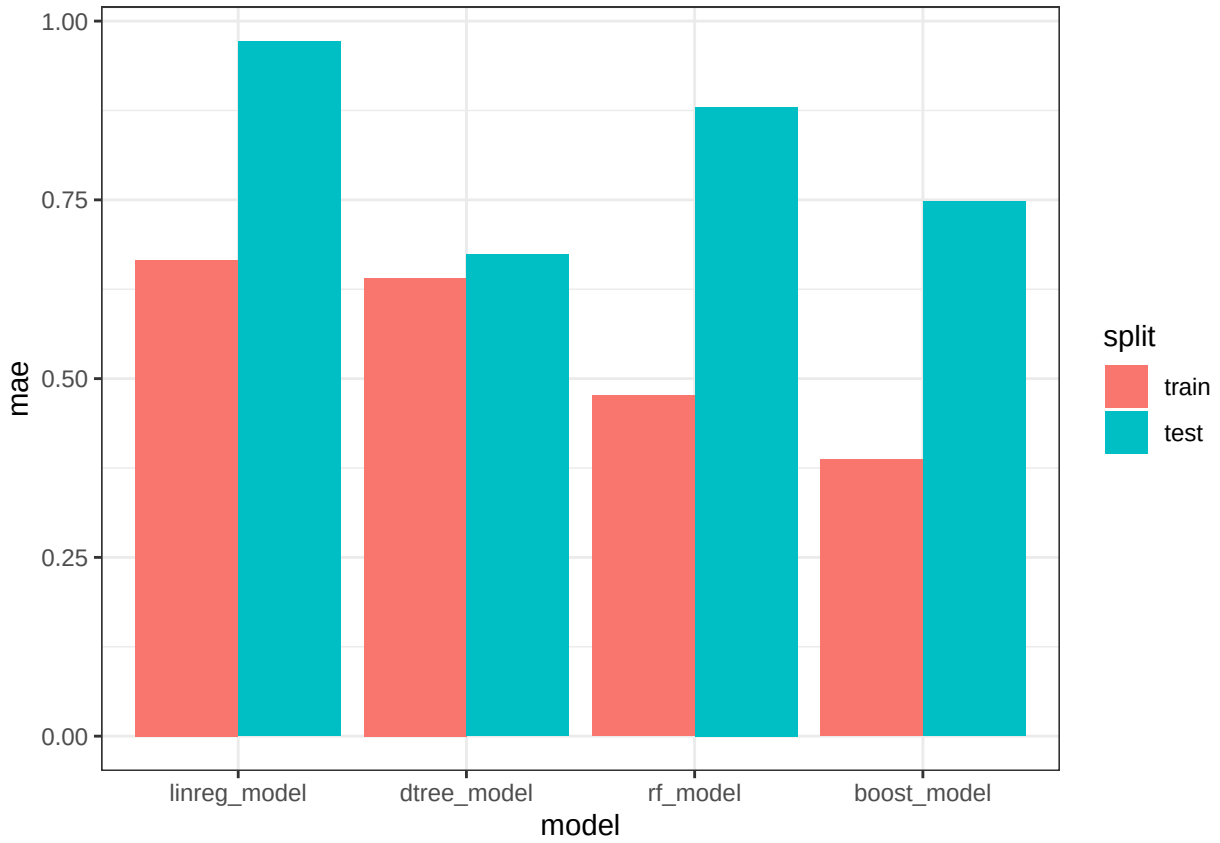



```
# Citation: Prof. Arnold (HW 7)
```

The decision tree model predicts too low for finals events and too high for non finals and semifinals events. The gradient boosted tree model with the added variables predicts slightly too high for almost all of the values.

Finally, we will quantify the performance of all of our models.

```
ms2_predictions <- bind_rows(
  linreg_model = augment(lm2_fit, new_data = eval_dataset),
  dtree_model = augment(dt2_fit, new_data = eval_dataset),
  rf_model = augment(rf2_fit, new_data = eval_dataset),
  boost_model = augment(bt2_fit, eval_dataset),
  .id = "model"
) %>% mutate(model = as_factor(model))

ms2_predictions %>%
  group_by(model, split) %>%
  mae(truth = result, estimate = .pred) %>%
  mutate(mae = .estimate) %>%
  ggplot(aes(x = model, y = mae, fill = split)) +
    geom_col(position = "dodge")
```




```
# Citation: Prof. Arnold (HW 7)
```

This bar graph shows that for the boost model was the best for the training set, and the decision tree model was the best for the test data. However, for both the boosted and random forest model. For all four models, there is over fitting that is larger than it was for the models prior to adding in new variables. The decision tree has the least over fitting.

Summary

Limitations and Social / Ethical Considerations

The largest limitation of our data analyses was the small size of our data set. While 82 races over a life time for a human is a lot for one individual especially for one running event, 82 observations does not make a large data set. Our performance analyses indicate that our models tended to over fit on the training data, leading to less accurate predictions on unseen data. In addition to this problem, Wayde van Niekerk's injury during 2018 to 2019 created a "gap" in the data that might not have been present in the analyses of another athlete. Though, ideally more track racers would be analyzed, it would take a significant amount of time to gather and format the data necessary to accomplish this. We decided to put our effort towards doing a very deep analysis on Wayde's time and seeing how the different factors affected our predictions.

Future Directions

..... In relation to the Manifesto for Data Practices, the principles that would help extend on our main question would be first and fourth principle. The first principle states, "Use data to improve life for our users, customers, organizations, and communities". The fourth states, "Prioritize the continuous collection and availability of discussions and metadata". In Data-Driven Track & Field/Cross Country, Doug Fenstermac+her eloquently argues, "With the rise of big data, data science, and machine learning, sports organizations have

began collecting performance data to inform their programs, and drive their organization forward. It's about time track & field did the same". In his blog post, Fenstermacher points out how that many aspects of track and field are measurable but that athletics entities often do not record or do not make such data publicly available. As such, we would also argue that these entities (including World Athletics) presently do not properly prioritize the continuous collection of data or consider how the data science community could contribute to fans, athletes, coaches, officials, and the sport of track and field as a whole. ... EXTEND UPON

Citations

The History Of Track And Field by Jessica Todd

Ancient Greek Athletes Who Defined the Olympic Games by Patricia Claus

World Athletics Championships in Eugene Drew More Than 18 Million Viewers Across NBC Sports Platforms by Kristi Turnquist

THANK YOU FOR A WONDERFUL SEMESTER!